UNIVERSITAT DE LES ILLES BALEARS

FACULTAD DE PSICOLOGÍA

TESIS DOCTORAL

Redes Neuronales Artificiales aplicadas al Análisis de Datos

JUAN JOSÉ MONTAÑO MORENO

DIRECTOR: DR. ALFONSO PALMER POL

PALMA DE MALLORCA, 2002

Este trabajo está dedicado a Ana

Agradecimientos

Quiero manifestar mi agradecimiento a mi maestro y director de tesis, Dr. Alfonso Palmer Pol, no sólo por su impecable labor de dirección sino también por sus continuas enseñanzas y apoyo en la elaboración de este trabajo. Quisiera también expresar mi gratitud a mis compañeros Alberto Sesé, Berta Cajal, Noelia Llorens y Rafael Jiménez, por su respaldo personal y amistad incondicional manifestada en el día a día. Igualmente, expreso mi agradecimiento a Carlos Fernández por su inestimable colaboración en la creación del programa informático *Sensitivity Neural Network 1.0*. Finalmente, pero no por ello menos importante, agradezco el apoyo siempre encontrado en mi familia y en mis amigos. A todos, muchísimas gracias.

Mientras los filósofos discuten si es posible o no la inteligencia artificial, los investigadores la construyen

C. Frabetti

ÍNDICE

PRÓLOGO

En los últimos años se ha consolidado un nuevo campo dentro de las ciencias de la computación que abarcaría un conjunto de metodologías caracterizadas por su inspiración en los sistemas biológicos para resolver problemas relacionados con el mundo real (reconocimiento de formas, toma de decisiones, etc.), ofreciendo soluciones robustas y de fácil implementación. Esta nueva forma de procesamiento de la información ha sido denominada Computación *Soft*, para distinguirla del enfoque algorítmico tradicional determinado por el binomio lógica booleana-arquitectura Von Neumann que, en este caso, sería la Computación *Hard*. Este conjunto de metodologías emergentes comprende la lógica borrosa, las redes neuronales, el razonamiento aproximado, los algoritmos genéticos, la teoría del caos y la teoría del aprendizaje.

De entre estas metodologías, las Redes Neuronales Artificiales son las que actualmente están causando un mayor impacto, debido a su extraordinaria aplicabilidad práctica. Recientemente esta tecnología ha captado la atención de los profesionales dedicados a la estadística y al análisis de datos, los cuales comienzan a incorporar las redes neuronales al conjunto de herramientas estadísticas orientadas a la clasificación de patrones y la estimación de variables continuas.

La presente tesis describe tres líneas de investigación desarrolladas en los últimos cinco años en torno a la aplicación de las redes neuronales artificiales en el ámbito del análisis de datos. Los campos de aplicación tratados son: el análisis de datos aplicado a conductas adictivas, el análisis de supervivencia, y el estudio del efecto de las variables de entrada en una red neuronal.

Como fruto de esta labor investigadora desarrollada por nuestro equipo, se han publicado nueve artículos en diversas revistas científicas, se han presentado seis trabajos en tres congresos de Metodología y, finalmente, se ha creado un programa simulador de redes neuronales. Esta tesis trata de aglutinar los logros alcanzados en este conjunto de trabajos, en los que se pone de manifiesto la utilidad de las redes neuronales en el ámbito del análisis de datos.

Juan José Montaño Moreno
Universidad de las Islas Baleares, Facultad de Psicología
Palma de Mallorca, Junio de 2002

1. Introducción

1.1. Redes neuronales artificiales (RNA). Concepto y evolución histórica.

Las Redes Neuronales Artificiales (RNA) o sistemas conexionistas son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. Consisten en un conjunto de elementos simples de procesamiento llamados nodos o neuronas conectadas entre sí por conexiones que tienen un valor numérico modificable llamado peso.

La actividad que una unidad de procesamiento o neurona artificial realiza en un sistema de este tipo es simple. Normalmente, consiste en sumar los valores de las entradas (*inputs*) que recibe de otras unidades conectadas a ella, comparar esta cantidad con el valor umbral y, si lo iguala o supera, enviar activación o salida (*output*) a las unidades a las que esté conectada. Tanto las entradas que la unidad recibe como las salidas que envía dependen a su vez del peso o fuerza de las conexiones por las cuales se realizan dichas operaciones.

La arquitectura de procesamiento de la información de los sistemas de RNA se distingue de la arquitectura convencional Von Neumann (fundamento de la mayor parte de los ordenadores existentes) en una serie de aspectos fundamentales.

En primer lugar, el procesamiento de la información de un modelo Von Neumann es secuencial, esto es, una unidad o procesador central se encarga de realizar una tras otra determinadas transformaciones de expresiones binarias almacenadas en la memoria del ordenador. Estas transformaciones son realizadas de acuerdo con una serie de instrucciones (algoritmo, programa), también almacenadas en la memoria. La operación básica de un sistema de este tipo sería: localización de una expresión en la memoria, traslado de dicha expresión a la unidad de procesamiento, transformación de la expresión y colocación de la nueva expresión en otro compartimento de la memoria. Por su parte, el procesamiento en un sistema conexionista no es secuencial sino paralelo, esto es, muchas unidades de procesamiento pueden estar funcionando simultáneamente.

En segundo lugar, un rasgo fundamental de una arquitectura Von Neumann es el carácter discreto de su memoria, que está compuesta por un gran número de ubicaciones físicas o compartimentos independientes donde se almacenan en código digital tanto las instrucciones (operaciones a realizar) como los datos o números que el ordenador va a utilizar en sus operaciones. En redes neuronales, en cambio, la información que posee

un sistema no está localizada o almacenada en compartimentos discretos, sino que está distribuida a lo largo de los parámetros del sistema. Los parámetros que definen el "conocimiento" que una red neuronal posee en un momento dado son sus conexiones y el estado de activación de sus unidades de procesamiento. En un sistema conexionista las expresiones lingüísticas o simbólicas no existen como tales. Serían el resultado emergente de la interacción de muchas unidades en un nivel subsimbólico.

Un sistema de procesamiento distribuido en paralelo presenta una serie de ventajas frente a un modelo convencional Von Neumann. Por un lado, tenemos la resistencia al funcionamiento defectuoso de una pequeña parte del sistema. En un modelo conexionista, cada unidad lleva a cabo una computación simple. La fiabilidad de la computación total que el sistema realiza depende de la interacción paralela de un gran número de unidades y, consecuentemente, en la mayoría de casos, el sistema puede continuar su funcionamiento normal, aunque una pequeña parte del mismo haya resultado dañada. En los sistemas convencionales, en cambio, un defecto en un solo paso de una larga cadena de operaciones puede echar a perder la totalidad de la computación. Por otro lado, un modelo conexionista es capaz, en ciertas circunstancias, de reconocer un objeto a pesar de que sólo se le presente como entrada una parte del mismo, o a pesar de que la imagen del objeto esté distorsionada. En cambio, en un sistema convencional el objeto presentado debe corresponderse con una determinada información almacenada en memoria, de lo contrario, no es capaz de reconocer el objeto.

Por último, un sistema de RNA no se programa para realizar una determinada tarea a diferencia de una arquitectura Von Neumann, sino que es "entrenado" a tal efecto. Consideremos un ejemplo típico de aprendizaje o formación de conceptos en la estructura de una RNA. Supongamos que presentamos a la red dos tipos de objetos, por ejemplo la letra A y la letra E con distintos tamaños y en distintas posiciones. En el aprendizaje de la red neuronal se consigue, tras un número elevado de presentaciones de los diferentes objetos y consiguiente ajuste o modificación de las conexiones del sistema, que la red distinga entre As y Es, sea cual fuere su tamaño y posición en la pantalla. Para ello, podríamos entrenar la red neuronal para que proporcionase como salida el valor 1 cada vez que se presente una A y el valor 0 en caso de que se presente una E. El aprendizaje en una RNA es un proceso de ajuste o modificación de los valores o pesos de las conexiones, "hasta que la conducta del sistema acaba por reproducir las

propiedades estadísticas de sus entradas" (Fodor y Pylyshyn, 1988, p. 30). En nuestro ejemplo, podríamos decir que la red ha "aprendido" el concepto de letra A y letra E sin poseer reglas concretas para el reconocimiento de dichas figuras, sin poseer un programa explícito de instrucciones para su reconocimiento.

Por tanto, para entrenar a un sistema conexionista en la realización de una determinada clasificación es necesario realizar dos operaciones. Primero, hay que seleccionar una muestra representativa con respecto a dicha clasificación, de pares de entradas y sus correspondientes salidas. Segundo, es necesario un algoritmo o regla para ajustar los valores modificables de las conexiones entre las unidades en un proceso iterativo de presentación de entradas, observación de salidas y modificación de las conexiones.

Las RNA constituyen una línea de investigación en Inteligencia Artificial (IA), la cual tiene como objetivo primario la construcción de máquinas inteligentes (Grimson y Patil, 1987). Los orígenes de la IA hay que buscarlos en el movimiento científico de la cibernética de los años cuarenta y cincuenta. Este movimiento científico se articuló en torno a la idea de que el funcionamiento de muchos sistemas, vivos o artificiales, puede ser captado mejor por modelos basados en la transferencia de información que por modelos basados en la transferencia de energía. La cibernética se propuso estudiar los elementos comunes entre el funcionamiento de máquinas automáticas y el del sistema nervioso humano (los procesos de control y comunicación en el animal y en la máquina). Este problema fue abordado en un esfuerzo interdisciplinar, en el que intervinieron investigadores procedentes de áreas como matemáticas, ingeniería electrónica, fisiología y neurociencia, lógica formal, ciencias de la computación y psicología.

Una importante característica de la cibernética fue la proliferación de distintas perspectivas en torno al problema de las relaciones entre cerebro y máquina. En la segunda mitad de la década de los cincuenta comenzaron a destacar dos de entre estas perspectivas: la IA basada en el procesamiento simbólico, y la investigación en redes neuronales.

La IA simbólica se basó en la expansión del uso de los ordenadores desde el área de aplicación del cálculo numérico a tareas simbólicas, esto es, al procesamiento de elementos que representan palabras, proposiciones u otras entidades conceptuales. Estos sistemas de IA se basan en las expresiones simbólicas que contienen y en la posibilidad

de manipular y transformar dichas expresiones de una manera sensible a la estructura lógico-sintáctica de las mismas. Las estructuras representacionales que contiene un sistema de este tipo son manipuladas y transformadas de acuerdo con ciertas reglas y estrategias (algoritmos y reglas heurísticas), y la expresión resultante es la solución de un determinado problema. En un sistema de este tipo, el procesamiento de la información tiene lugar en el nivel simbólico o representacional y no en el nivel neurobiológico. Los sistemas de IA simbólica simulan procesos mentales y cognitivos humanos por medio de programas ejecutados por un ordenador del tipo Von Neumann. Entre los investigadores más importantes de esta primera época de investigación en este paradigma se puede destacar a John McCarthy, Allen Newell, Herbert Simon y Marvin Minsky (Olazarán, 1993).

Paralelamente, en la segunda mitad de los años 50, algunos investigadores comenzaron a desarrollar una perspectiva diferente en la construcción de máquinas inteligentes: la perspectiva de las RNA o sistemas conexionistas. Esta perspectiva no perseguía la modelación de redes neuronales fisiológicas, sino la construcción de máquinas inteligentes empleando arquitecturas computacionales de cierta semejanza con las redes neuronales del cerebro. Como antecedentes más directos a este grupo de investigadores, cabe destacar las aportaciones, por un lado, de Warren McCulloch y Walter Pitts y, por otro lado, de Donald Hebb.

McCulloch y Pitts (1943) presentaron la estructura y funcionamiento de la unidad elemental de procesamiento de una red conexionista. La neurona de McCulloch-Pitts (ver figura 1), como actualmente se conoce, tiene un funcionamiento muy sencillo: si la suma de entradas excitatorias supera el umbral de activación de la unidad, y además no hay una entrada inhibitoria, la neurona se activa y emite respuesta (representada por el valor 1); en caso contrario, la neurona no se activa (valor 0 que indica la ausencia de respuesta).

Combinando varias neuronas de este tipo con los adecuados umbrales de respuesta, se puede construir una red que compute cualquier función lógica finita.

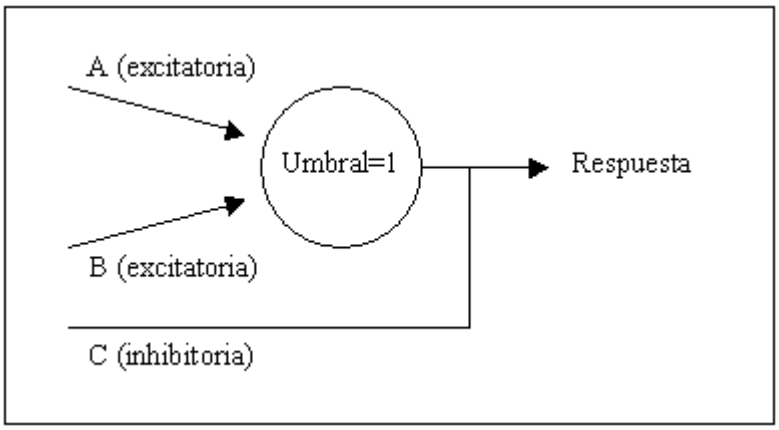


Figura 1. Neurona de McCulloch-Pitts.

Hebb (1949) postuló un sencillo pero potente mecanismo de regulación de las conexiones neuronales, que constituyó la base de las reglas de aprendizaje que más tarde se desarrollarían. La regla de Hebb, en su versión más elemental, se expresa como sigue: "*Cuando un axón de una célula A está bastante cerca para excitar a una célula B y repetida o persistentemente dispara, entonces se produce algún proceso de desarrollo o cambio metabólico de tal forma que la eficiencia del disparo de A hacia B aumenta*" (Hebb, 1949, p. 42). La propuesta de Hebb es de especial relevancia porque indica que la información necesaria para modificar el valor de una conexión se encuentra localmente disponible a ambos lados de la conexión. En la actualidad existe un gran número de redes neuronales cuyo aprendizaje está basado en la regla de Hebb como las conocidas redes de Hopfield (1982) y algunos modelos de red propuestos por Kohonen (1977).

La evolución de la investigación en redes neuronales desde los años 50 a nuestros días ha estado condicionada por dos grandes acontecimientos: el abandono de esta línea de investigación en la segunda mitad de los 60 debido a las limitaciones observadas en la red Perceptrón simple y la emergencia del conexionismo en la segunda mitad de los 80 como paradigma aceptado en IA, gracias, entre otros avances, a la aparición de un algoritmo, denominado *backpropagation error* (propagación del error hacia atrás) o simplemente *backpropagation*, que permite modificar las conexiones de arquitecturas multiestrato.

En el primer período de la investigación en redes neuronales, entre mediados de los 50 y mediados de los 60, una cantidad importante de científicos, ingenieros y grupos de investigación dedicaron importantes esfuerzos a la construcción y experimentación de

estos sistemas. Entre los grupos más importantes se podrían destacar el grupo de F. Rosenblatt en la Universidad de Cornell (Nueva York), el grupo de C. Rosen en el Instituto de Investigación de Stanford (California), y el grupo de B. Widrow en el Departamento de Ingeniería Electrónica de la Universidad de Stanford.

En este período se produjeron importantes contribuciones científicas. Una de las más importantes fue el trabajo de los grupos de Rosenblatt y Widrow con sistemas conexionistas de único estrato o capa (RNA que solo tienen un estrato de conexiones modificables). La red diseñada por Rosenblatt (1958), denominada Perceptrón, es un sistema de este tipo (ver figura 2). A pesar de tener dos estratos de conexiones, sólo uno de ellos está compuesto de conexiones modificables. La capa de entrada o retina consiste en un conjunto de unidades de entrada binarias conectadas por conexiones con valor fijo con las unidades de la capa de asociación o de predicados. La última capa es la de respuesta o decisión, cuya única unidad, con salida binaria, es la que tiene conexiones modificables con los predicados de la capa anterior.

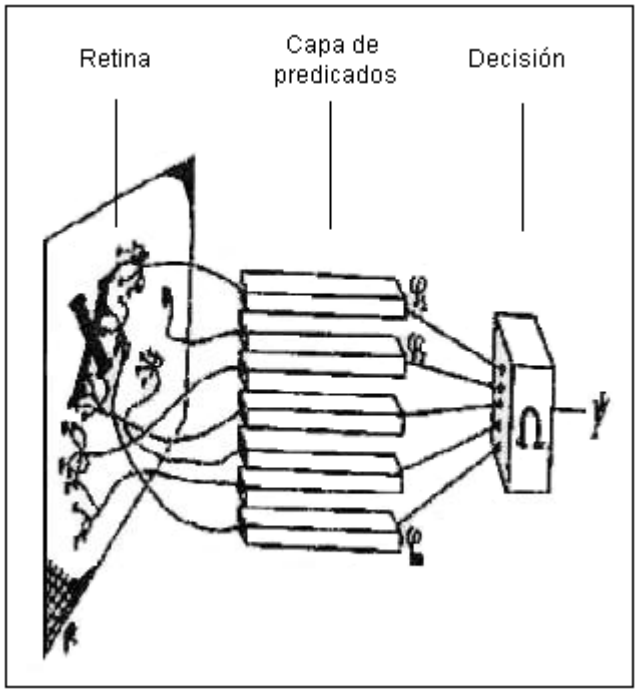


Figura 2. El Perceptrón de Rosenblatt.

El teorema de convergencia de la regla de aprendizaje del Perceptrón desarrollado por Rosenblatt establecía que, si los parámetros o pesos del sistema eran capaces de realizar una determinada clasificación, el sistema acabaría aprendiéndola en un número finito de

pasos, si se modificaban las conexiones de acuerdo con dicha regla de aprendizaje (Fausett, 1994). Más concretamente, la regla de aprendizaje del Perceptrón es un algoritmo de los denominados supervisado por corrección de errores y consiste en ir ajustando de forma iterativa los pesos en proporción a la diferencia existente entre la salida actual de la red y la salida deseada, con el objetivo de minimizar el error actual de la red.

La polémica suscitada entre científicos favorables y contrarios al conexionismo fue aumentando en la segunda mitad de los 50 conforme el trabajo de Rosenblatt fue adquiriendo notoriedad. Rosenblatt, un psicólogo de la Universidad de Cornell (Ithaca, Nueva York), fue la figura central del conexionismo de los años cincuenta y sesenta. El Perceptrón, una máquina conexionista diseñada y estudiada teóricamente por Rosenblatt, construida por un grupo de ingenieros del Laboratorio de Aeronáutica de Cornell (CAL, Ithaca, Nueva York) y financiada por la Oficina de Investigación Naval del Ejército de los Estados Unidos (ONR, *Office of Naval Research*), fue una de las contribuciones científicas y tecnológicas más importantes de la primera fase del conexionismo.

Otra importante contribución científica es la aportada por Widrow y Hoff en 1960. Estos autores propusieron un nuevo tipo de unidad de procesamiento, con estructura similar a la del Perceptrón pero con un mecanismo de aprendizaje diferente que permitía también la entrada de información de tipo continuo: la neurona ADALINE (ADAptative LINear Elements) (Widrow y Hoff, 1960). La innovación de esta tipología de neurona se halla en su mecanismo de aprendizaje denominado regla delta o regla de Widrow-Hoff, que introduce el concepto de reducción del gradiente del error. La deducción de la regla delta se puede expresar de la siguiente forma: teniendo en cuenta que E_p (el error que comete la red para un determinado patrón p), es función de todos los pesos de la red, el gradiente de E_p es un vector igual a la derivada parcial de E_p respecto a cada uno de los pesos. El gradiente toma la dirección del incremento más rápido en E_p; la dirección opuesta toma el decremento más rápido en el error. Por tanto, el error puede reducirse iterativamente ajustando cada peso w_i en la dirección $-\frac{\partial E_p}{\partial w_i}$ (Widrow y Hoff, 1960). Como veremos más adelante, la regla delta basada en la reducción del

gradiente del error es la precursora del algoritmo *backpropagation* aplicado a redes de múltiples estratos.

Sin embargo, los primeros sistemas conexionistas tenían importantes limitaciones técnicas. Una de las más importantes es que una neurona tipo Perceptrón solamente permite discriminar entre dos clases linealmente separables, es decir, cuyas regiones de decisión pueden ser separadas mediante una única recta o hiperplano (dependiendo del número de entradas). Otra importante limitación era la carencia de técnicas para la modificación de conexiones en sistemas de múltiples estratos. Este problema se puede ilustrar con las conocidas funciones OR y OR-Exclusiva (XOR). En el caso de la función OR, un Perceptrón de una sola capa de conexiones modificables permite solucionar esta función debido a que el problema es linealmente separable (ver figura 3 izquierda). En cambio, en el caso de la función OR-Exclusiva, un Perceptrón de este tipo no permite solucionar esta función debido a que no existe ninguna recta que separe los patrones de una clase de los de la otra. Para ello es necesario que se introduzca una capa intermedia compuesta por dos neuronas que determinen dos rectas en el plano (ver figura 3 derecha).

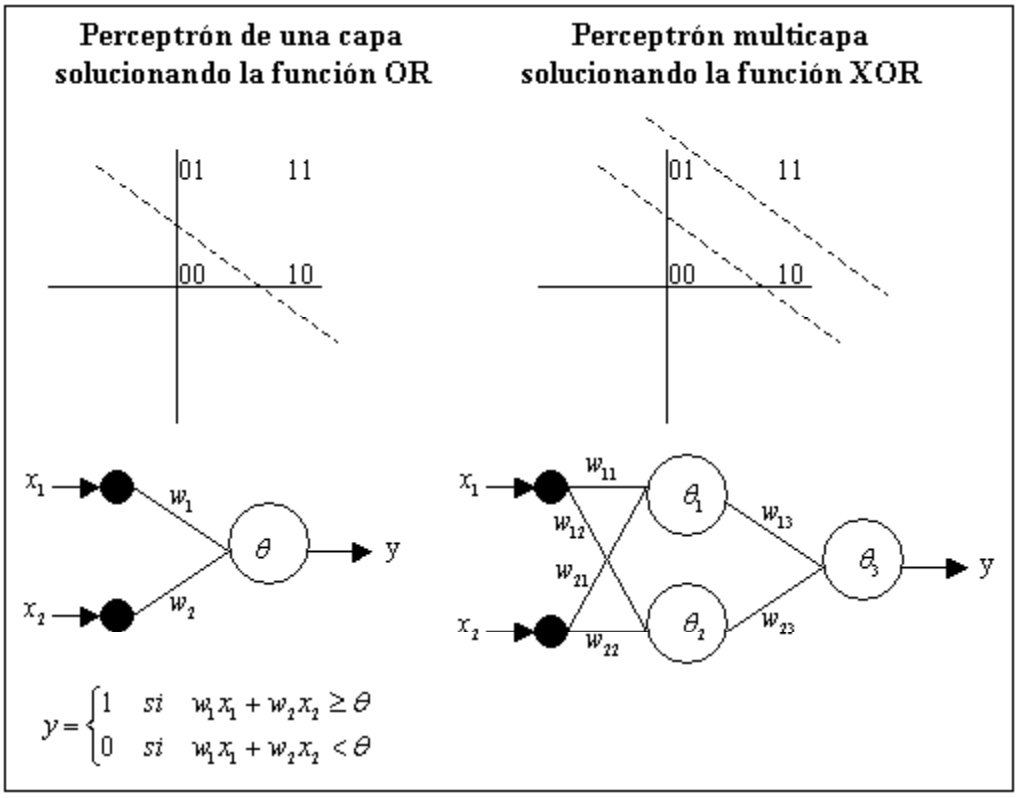


Figura 3. Perceptrones solucionando la función OR y la función XOR.

Los primeros investigadores conexionistas eran conscientes de que la falta de un algoritmo para la modificación de conexiones en sistemas de múltiples estratos limitaba considerablemente la capacidad de clasificación de objetos de los sistemas conexionistas, y de que un sistema de múltiples estratos era capaz de realizar cualquier clasificación.

Estos investigadores se enfrentaban también a importantes problemas tecnológicos. Una de las limitaciones más claras de los ordenadores conexionistas de este primer período era su tamaño. El Perceptrón construido por los ingenieros colaboradores de Rosenblatt en CAL, que tenía tan sólo 512 conexiones modificables, ocupaba todo un pabellón de dicho centro. La razón de esto es que cada conexión era implantada utilizando un potenciómetro con motor de considerable tamaño. Implantar un Perceptrón con decenas de miles de conexiones modificables con esta tecnología era impracticable. Aunque los investigadores conexionistas intentaron otras alternativas, la tecnología "neuronal" estaba en claro declive que coincidía con el ocaso de los ordenadores analógicos y con el despegue de la tecnología de computación digital secuencial de tipo Von Neumann. Los avances en la tecnología Von Neumann benefició al paradigma de IA que desde un principio se basó en dicha tecnología: el paradigma simbólico. Por otro lado, la falta de afinidad entre el ordenador digital y el conexionismo, y la reducida potencia de los ordenadores digitales de aquella época hicieron que apenas se considerara la posibilidad de simular RNA en dichos ordenadores.

El declive del primer conexionismo sobrevino cuando Marvin Minsky y Seymour Papert, dos investigadores líderes de la IA simbólica del prestigioso Instituto de Tecnología de Massachusetts (MIT), publican en 1969 el libro *Perceptrons* (Minsky y Papert, 1969) donde se realizaba una contundente crítica a los modelos de Perceptrón propuestos por Rosenblatt. Las aportaciones principales del estudio de Minsky y Papert pueden agruparse en dos bloques. Por un lado, Minsky y Papert realizaron un estudio, muy elaborado desde un punto de vista matemático, de algunos de los problemas que presentaban las redes de único estrato. En concreto demostraron que el Perceptrón de una capa, actualmente denominado Perceptrón simple, era incapaz de diferenciar entre entradas en distintas partes de la pantalla (triángulo a la derecha, triángulo a la izquierda), ni entre figuras en distintas posiciones de rotación. Tampoco era capaz de computar con efectividad funciones matemáticas como la paridad (dada una cantidad de puntos activos en la retina, reconocer si es un número par o impar), la función

topológica de la conectividad (reconocer una figura como una totalidad separada del fondo) y en general funciones no lineales como la mencionada función OR-Exclusiva. Por otro lado, el segundo conjunto de resultados del estudio de Minsky y Papert es el referido a las RNA de múltiples estratos. En este caso dedicaron mucho menos espacio a este problema en su libro, concluyendo que "*el estudio de las versiones de múltiples estratos es estéril*" (Minsky y Papert, 1969, p. 232) alegando que sería muy improbable obtener una regla de aprendizaje aplicada a este tipo de arquitecturas.

Según Olazarán (1993), la polémica suscitada en torno a los primeros modelos de red neuronal entre simbolismo y conexionismo hay que situarla en un contexto social, en el que ambos grupos competían por erigirse como paradigma dominante en el campo de la IA, y también por conseguir el apoyo económico de agencias militares como ONR y, sobretodo, DARPA (*Defense Advanced Research Projects Agency*, la Agencia de Proyectos de Investigación Avanzados del Ministerio de Defensa de los Estados Unidos). Los investigadores de IA simbólica vieron al conexionismo como una amenaza directa para sus intereses, y se opusieron a que las agencias militares apoyaran económicamente proyectos de envergadura en RNA.

La polémica de los años setenta entre el simbolismo y el conexionismo terminó con la aceptación por la gran mayoría de los científicos de la IA, del paradigma simbólico como línea de investigación más viable. La credibilidad que la élite de IA simbólica (Herbert Simon, Allen Newell, Marvin Minsky y John McCarthy) consiguió tanto dentro de la comunidad científica (estos investigadores dominaron la disciplina) como fuera de ella (apoyo económico de DARPA) es un indicativo de la posición favorable en la que estos investigadores quedaron cuando la polémica sobre el Perceptrón se dio por terminada. Ante la situación de crisis, algunos de los principales grupos de RNA abandonaron su investigación. El grupo de Widrow comenzó a aplicar sus técnicas y sistemas de RNA a la ingeniería de las telecomunicaciones, y el grupo de Rosen comenzó un proyecto para la construcción de un robot móvil dentro del paradigma simbólico de IA. Rosenblatt y algunos otros investigadores, en cambio, continuaron con sus investigaciones en RNA. De hecho, la mayoría de los actuales líderes en el campo de las RNA comenzaron a publicar sus trabajos durante la década de los 70. Este es el caso de investigadores como James Anderson, Teuvo Kohonen, Christoph Von Der Malsburg, Kunihiko Fukushima, Stephen Grossberg y Gail Carpenter que pasamos a comentar brevemente.

Anderson desarrolló un asociador lineal de patrones que posteriormente perfeccionó en el modelo BSB (*Brain-State-in-a-Box*) (Anderson, Silverstein, Ritz y Jones, 1977). Simultáneamente, en Finlandia, Kohonen desarrolló un modelo similar al de Anderson (Kohonen, 1977); años más tarde, crearía un modelo topográfico con aprendizaje autoorganizado en el que las unidades se distribuyen según el tipo de entrada al que responden (Kohonen, 1982). Este modelo topográfico, comúnmente denominado mapa autoorganizado de Kohonen, es una de las redes neuronales más ampliamente utilizadas en la actualidad.

En Alemania, Von Der Malsburg (1973) desarrolló un detallado modelo de la emergencia en la corteza visual primaria de columnas de neuronas que responden a la orientación de los objetos. En Japón, Fukushima desarrolló el Cognitrón (Fukushima, 1975), un modelo de red neuronal autoorganizada para el reconocimiento de patrones visuales. Posteriormente, presentó la red Neocognitrón (Fukushima, 1980, 1988; Fukushima, Miyake e Ito, 1983) que permitía superar las limitaciones del primitivo Cognitrón.

Por su parte, Grossberg ha sido uno de los autores más prolíficos en este campo. Klimasauskas (1989) lista 146 publicaciones en las que interviene Grossberg entre 1967 y 1988. Estudió los mecanismos de la percepción y la memoria. Grossberg realizó en 1967 una red, Avalancha, que consistía en elementos discretos con actividad que varía con el tiempo que satisface ecuaciones diferenciales continuas, para resolver actividades tales como reconocimiento continuo del habla y aprendizaje del movimiento de los brazos de un robot (Grossberg, 1982). Sin embargo, la contribución más importante de Grossberg es la Teoría de Resonancia Adaptativa (ART), desarrollada en colaboración con Carpenter (Carpenter y Grossberg, 1985, 1987a, 1987b, 1990). La ART se aplica a modelos con aprendizaje competitivo (denominados ART para la versión no supervisada y ARTMAP para la versión supervisada) en los cuales cuando se presenta cierta información de entrada sólo una de las neuronas de salida de la red se activa alcanzando su valor de respuesta máximo después de competir con las demás neuronas.

En la década de los años 80 coincidieron una serie de acontecimientos que jugaron un papel relevante en la reemergencia del conexionismo. En esos momentos, la IA simbólica se encontraba en una fase de comercialización tras el anuncio del Programa de la Quinta Generación de Ordenadores por parte del gobierno japonés y el desarrollo

de los sistemas expertos. No obstante, a pesar del éxito de estos sistemas en ciertas áreas de aplicación, un número creciente de investigadores comenzaba a ser consciente de las limitaciones de los sistemas simbólicos ante ciertas tareas --denominadas del mundo real--, como el reconocimiento de objetos, el reconocimiento de lenguaje hablado y el razonamiento de sentido común. Conforme avanzaba la década de los ochenta, estas limitaciones condujeron a investigadores procedentes de diversas áreas a realizar aportaciones alternativas a las propuestas por la IA simbólica.

En este sentido, uno de los casos más paradigmáticos es el del físico John Hopfield, considerado como uno de los impulsores más importantes del nuevo conexionismo. Hopfield publicó en 1982 un importante artículo en la Academia Nacional de las Ciencias (Hopfield, 1982). Este escrito claro y conciso tuvo un importante impacto en el campo por varias razones. En primer lugar, Hopfield era un conocido físico con conexiones institucionales importantes. Su interés y trabajo en redes neuronales legitimó el campo para la comunidad científica. En segundo lugar, impulsó la implementación de los modelos de red mediante dispositivos electrónicos utilizando tecnología VLSI (Muy Alta Escala de Integración). En tercer lugar, Hopfield sugirió una estrecha relación entre los sistemas físicos y las redes neuronales. El concepto clave de las redes propuestas por Hopfield es que considera la fase de ajuste de las conexiones como una búsqueda de valores mínimos en unos paisajes de energía. Según esta idea, cada combinación de pesos de las conexiones de la red tiene asociada una energía, que resulta de evaluar las restricciones determinadas por los datos de entrada y el resultado producido por la red. El intercambio de información entre unidades se mantiene hasta que la entrada y la salida de cada unidad sean iguales, es decir, en términos de Hopfield se ha llegado a un estado de equilibrio energético. A diferencia de las redes Perceptrón y ADALINE, las redes utilizadas por Hopfield poseen una arquitectura monocapa cuyas conexiones son modificadas a partir de un algoritmo de aprendizaje basado en la regla de Hebb. Las redes de Hopfield han sido empleadas como memorias autoasociativas, principalmente para el reconocimiento de patrones.

El modelo de Hopfield fue posteriormente desarrollado por Hinton y Sejnowski, dos de los más importantes miembros del grupo de investigación PDP (*Parallel Distributed Processing*) (Universidad de San Diego, California), en su sistema denominado "máquina de Boltzmann" (Ackley, Hinton y Sejnowski, 1985). El algoritmo para la modificación de conexiones del sistema de múltiples estratos de Hinton y Sejnowski fue

una de las aportaciones más importantes de la primera fase de la reemergencia del conexionismo de los 80. Era la primera vez que un algoritmo de este tipo encontraba una aceptación considerable en la comunidad científica.

Sin embargo, la contribución más importante en la reemergencia del conexionismo en los años ochenta fue la técnica *backpropagation* desarrollada por Rumelhart, Hinton y Williams, representantes del grupo PDP. Realmente, esta técnica fue desarrollada inicialmente por Paul Werbos (1974) a mediados de los 70, y después independientemente redescubierta por varios grupos de investigadores (Le Cun, 1985; Parker, 1985; Rumelhart, Hinton y Williams, 1986). Es, por tanto, un caso de "descubrimiento múltiple". Sin embargo, en general se reconoce que fue la versión del grupo PDP la que desató el interés en RNA a mediados de los ochenta y consiguió finalmente forzar la revisión del consenso contrario al conexionismo.

El algoritmo *backpropagation* también recibe el nombre de regla delta generalizada o método de gradiente decreciente, debido a que supone una extensión de la regla propuesta por Widrow y Hoff en 1960 (regla delta) a redes con capas intermedias (ver figura 4). Este tipo de arquitectura recibe el nombre genérico de Perceptrón Multicapa o MLP (*Multilayer Perceptron*). Rosenblatt ya tuvo la idea de utilizar una técnica de este tipo a principios de los sesenta (Rosenblatt, 1962), aunque no pudo desarrollarla de un modo satisfactorio.

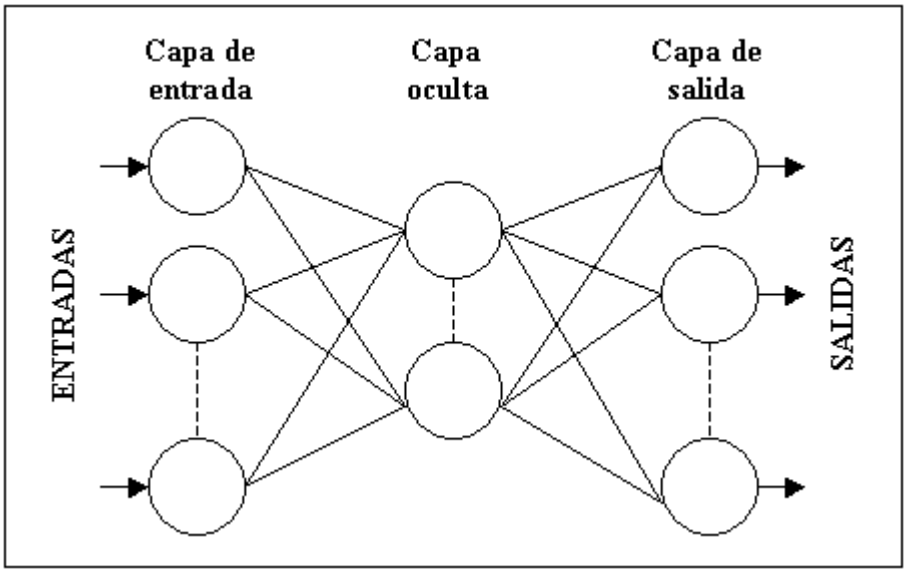


Figura 4. Arquitectura de un Perceptrón multicapa.

Como se comentó anteriormente, la falta de un algoritmo para la modificación de conexiones en sistemas de múltiples estratos limitaba considerablemente la capacidad de clasificación de objetos en los sistemas conexionistas de los años 60. En este sentido, el problema principal de la modificación de los valores de las conexiones en una red MLP es hallar el error cometido por las unidades de las capas intermedias. El error cometido por las unidades de salida es inmediatamente visible: es la diferencia entre la salida producida por dichas unidades y la salida que se desea que produzcan. El objetivo del algoritmo *backpropagation* es propagar los errores cometidos por las unidades de salida hacia atrás, ya que, en un sistema de este tipo, el error cometido por una unidad intermedia depende del error cometido por las unidades de salida a las que dicha unidad intermedia está conectada. Tras conocerse el error cometido por las unidades intermedias, pueden entonces modificarse las conexiones entre unidades de entrada y unidades intermedias. De forma similar a la regla delta, la base matemática del algoritmo *backpropagation* es la técnica de gradiente decreciente, basada en modificar los pesos en la dirección opuesta al gradiente, esto es $-\frac{\partial E_p}{\partial w_{ij}}$, en la dirección que determina el decremento más rápido del error.

Una novedad muy importante en el sistema de Rumelhart y sus colegas fue la introducción de funciones de activación continuas en todas las unidades de procesamiento en lugar de la clásica función “escalón” del Perceptrón simple de Rosenblatt. De hecho, el algoritmo *backpropagation* exige la utilización de funciones de activación continuas para poder realizar el cálculo de la derivada parcial del error con respecto a los pesos del modelo.

El proceso de acumulación de resultados e investigaciones y de esfuerzo organizacional por parte del grupo PDP, comenzó a hacer peligrar el consenso anticonexionista con el que terminó la polémica del Perceptrón. Los dos volúmenes PDP, considerados como la “biblia” del conexionismo, son el mayor exponente de este esfuerzo (Rumelhart, McClelland y el grupo de investigación PDP, 1986; McClelland, Rumelhart y el grupo de investigación PDP, 1986). El debate sobre el conexionismo se estaba reabriendo, y ésto hizo reaccionar de nuevo a los investigadores críticos con el conexionismo. La reacción fue encabezada, una vez más, por Minsky y Papert que, en el epílogo a la nueva edición de su libro *Perceptrons* (Minsky y Papert, 1988), criticaron contundentemente las afirmaciones de Rumelhart y sus colegas acerca de los sistemas

de múltiples estratos con el algoritmo *backpropagation*. Minsky y Papert no fueron los únicos en criticar al nuevo conexionismo con vehemencia. Otros científicos líderes en sus áreas de investigación, tales como Poggio (visión), Hillis (ordenadores paralelos) y Fodor y Pylyshyn (ciencia cognitiva), también realizaron críticas radicales al conexionismo (Olazarán, 1991). Sin embargo, esta vez la polémica no acabó con el abandono del conexionismo como ocurriera en la década de los 60.

En el artículo de Horgan (1994) se trata la persona de Marvin Minsky, comentándose algunas de sus opiniones actuales, como, por ejemplo, cómo poco a poco se ha ido apartando de la IA simbólica y su aprobación al actual desarrollo de las RNA.

Gracias al esfuerzo de movilización y acumulación científica y organizacional que el grupo de investigación PDP realizó a lo largo de la década de los ochenta, el conexionismo ha logrado en la actualidad diferenciarse como una especialidad científica aceptada, dentro del marco general de la IA. Este proceso ha culminado con el surgimiento, crecimiento e institucionalización de una comunidad científica diferenciada con su correspondiente sistema de comunicación y control especializado (publicaciones científicas, congresos, cursos de postgrado, institutos de investigación, programas y becas en las agencias que financian la investigación científica, etc.).

1.2. Estudio bibliométrico sobre RNA.

Como primera labor de investigación en el campo de las RNA, nos propusimos la creación de una base de datos que recopilase el mayor número posible de trabajos sobre RNA en el ámbito de la Psicología y el análisis de datos. También estabamos interesados en analizar trabajos pertenecientes a otros ámbitos (como medicina, biología, ingeniería, etc.), ya que podrían aportarnos nuevas ideas para posteriormente ser aplicadas en el campo de la Psicología y la Metodología.

Para la creación del fondo bibliográfico, fueron seleccionadas ocho bases de datos cuya descripción y resultados generales obtenidos se pueden consultar en la tabla 1. La selección de las bases de datos se realizó en función de su disponibilidad y de la adecuación de su contenido a los objetivos del estudio.

Tabla 1. Bases de datos consultadas (Cajal et al., 2001).

Base de datos	Descripción	Editor	Nº registros
Dissertation Abstracts	Recoge citas (con resumen) de aprox. un millón de tesis doctorales y "masters" desde 1861 de unas 500 universidades.	University Microfilms International	1251
Eric	Comprende citas (con resumen) de educación del *Educational Resources Information Center del US Department of Education*. Recoge las fuentes: RIE y CIJE. Contiene información desde 1966.	Dialog Information Services	137
ISBN	Base de datos sobre libros registrados en España desde 1972.	Agencia Española del ISBN	18
Library of Congress	Catálogo de la Librería del Congreso norteamericana. Mantiene un catálogo, accesible desde intenet, con las publicaciones incluidas en su registro informatizado desde 1968 (con más de 4,5 millones de registros).	Library of Congress	1277
Medline	Base de datos de la *US National Library of Medicine*. Incluye las citas (con resumen) de los artículos publicados en más de 3.000 revistas biomédicas, un 75% de las cuales están en lengua inglesa.	Cambridge Scientific Abstracts	2810
PsycLit	Base de datos de la *American Psychological Association*. Equivale a la publicación *Psychological Abstracts*. Indexa más de 1.300 revistas especializadas en psicología y ciencias del comportamiento. Recoge materiales relativos a psicología, psiquiatría, sociología, antropología, educación, etc. Contiene información desde 1974.	SilverPlatter Information, Inc.	2201
Sociofile	Base de datos que equivale a la publicación *Sociological Abstracts*. Incluye referencias (con resumen) sobre sociología aparecidas desde 1974. Incorpora la base de datos SOPODA que contiene información desde 1980.	SilverPlatter Information, Inc.	64
Teseo	Contiene información (cita y resumen) sobre tesis doctorales leídas en universidades españolas desde 1976.	Ministerio de Educación, Cultura y Deporte	133
			7891

Sobre la base de datos generada se aplicaron un conjunto de procedimientos derivados de la investigación bibliométrica y basados en análisis estadísticos descriptivos y sociométricos, que permitieron descubrir información valiosa acerca de la producción científica en el campo de las RNA como, por ejemplo, autores más productivos, revistas y editoriales dominantes, líneas de investigación actuales o utilización de RNA en las diferentes áreas de la Psicología y el análisis de datos, etc. Estos resultados fueron

publicados en la Revista de la Asociación Española de Metodología de las Ciencias del Comportamiento (AEMCCO) (Cajal et al., 2001) (ver apartado 2.3., pág. 149). El lector interesado en la metodología del análisis bibliométrico puede consultar los excelentes trabajos de Carpintero (1980), Méndez (1986), Sancho (1990), Alcain (1991), Romera (1992), Ferreiro (1993) y Amat (1994). A continuación, se describen los principales resultados obtenidos en el análisis bibliométrico que permitieron establecer las líneas de investigación que determinan la presente tesis.

1.2.1. Resultados generales.

El análisis de la evolución temporal de la productividad sobre la base de datos creada, nos permitió averiguar si el interés por las RNA ha crecido, ha declinado o se mantiene estable durante el período de tiempo observado. En este sentido, el gráfico 1 muestra la evolución temporal de la productividad desde 1980 hasta 1998.

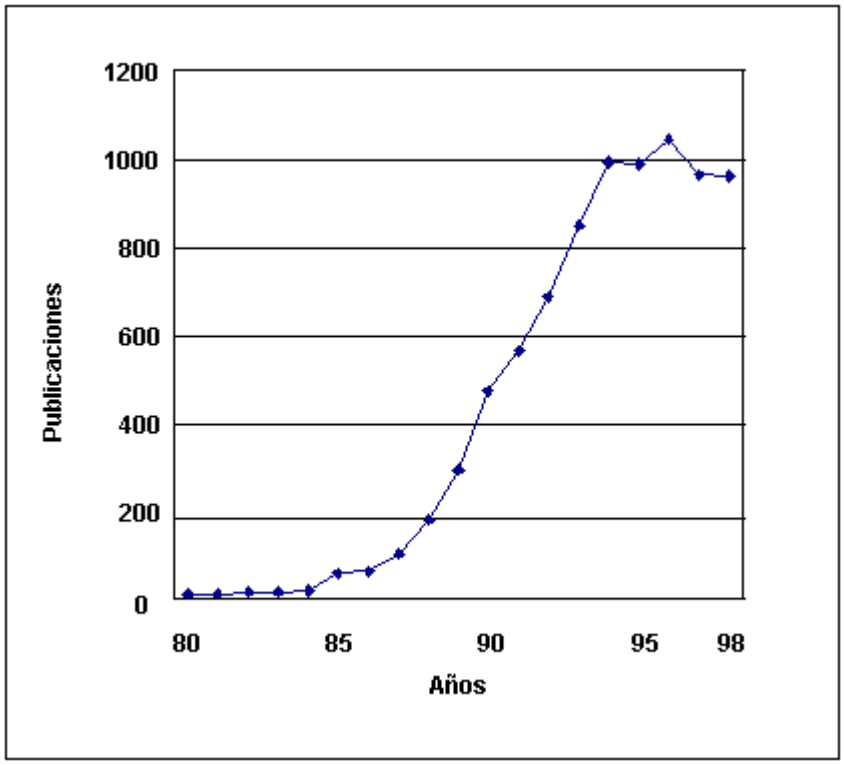


Gráfico 1. Evolución temporal de la productividad (Cajal et al., 2001).

Como se puede observar, el grado de producción o interés por las RNA es mínimo hasta aproximadamente la mitad de los años 80. A partir de esa fecha el interés comienza a

aumentar, primero de forma tímida, y a partir de 1990 de forma significativa alcanzando un pico de producción situado en el año 1996, que cuenta con 1.048 publicaciones. En el año 1994 se da la mayor producción de libros e informes técnicos, mientras que en los años 1995 y 1996 se da la mayor producción de tesis doctorales y artículos, respectivamente.

Sin duda, el creciente interés despertado por las RNA, que se manifiesta de forma palpable a principios de los 90, está relacionado con un acontecimiento fundamental en la historia de las RNA comentado al inicio, a saber: la publicación de *Parallel Distributed Processing* (Procesamiento Distribuido en Paralelo o PDP) (Rumelhart, McClelland y el grupo PDP, 1986; McClelland, Rumelhart y el grupo de investigación PDP, 1986), obra que se ha llegado a conocer como la "biblia" del nuevo paradigma conexionista donde se describe, entre otras cosas, el algoritmo *backpropagation* aplicado a redes MLP.

A continuación, todos los registros de la base de datos fueron clasificados en una de entre ocho materias o áreas temáticas. Esta labor clasificatoria no sólo permitió realizar labores de filtrado y discriminación de los registros en función de su temática, sino también analizar el grado de interés que los autores de RNA otorgan a las diferentes materias. A continuación se presentan las ocho categorías temáticas utilizadas junto con una breve descripción de su contenido:

- Algoritmos: presentación de esquemas de aprendizaje, algoritmos, arquitecturas, análisis de su rendimiento y presentación de métodos de optimización.
- Aplicaciones: aplicación práctica de las RNA en algún área de conocimiento: psicología, medicina, ingeniería, economía, etc.
- Comparación con otros modelos: comparación del rendimiento de las RNA con modelos estadísticos clásicos y modelos derivados de la Inteligencia Artificial simbólica.
- Epistemología: discusiones sobre filosofía de la mente y conexionismo.
- Hardware/Software: implementación en hardware de arquitecturas neuronales y presentación o evaluación de programas simuladores de RNA.
- Manual/Introducción: manuales de consulta y trabajos divulgativos o de introducción al campo de las RNA.
- Modelado de procesos: utilización de modelos conexionistas para el estudio y simulación de procesos fisiológicos (principalmente cerebrales) y cognitivos.

- Otros: trabajos muy generales sin ubicación específica.

Como se puede observar en el gráfico 2, el área temática que cuenta con más registros es el modelado de procesos (2.139 registros), principalmente fisiológicos y psicológicos, perfilándose como la línea de investigación predominante. También acumulan un elevado número de publicaciones las categorías: aplicaciones (2.057 registros), manual/introducciones (1.665 registros) y algoritmos (1.024 registros).

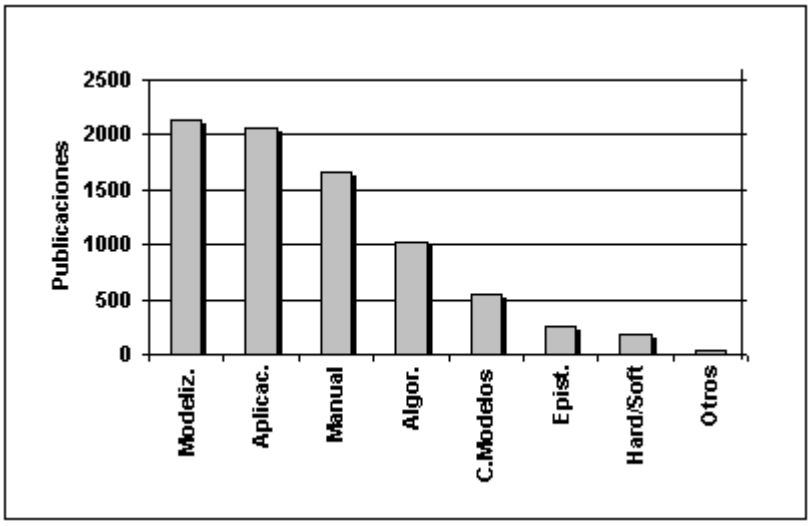


Gráfico 2. Distribución de las materias (Cajal et al., 2001).

Respecto a la cuestión de cuántos tipos de redes neuronales existen actualmente, se puede decir que se trata de un número inabarcable. Sin embargo, también se puede decir que del total hay aproximadamente 40 modelos que son bien conocidos por la comunidad de investigadores en RNA. A continuación, se presenta la tabla 2 con la clasificación de las RNA más conocidas en función del tipo de aprendizaje utilizado: supervisado o no supervisado:

Tabla 2. Clasificación de las RNA más conocidas.

1. Supervisado
1. Con conexiones feedforward
- Lineales
- Perceptrón (Rosenblatt, 1958)
- Adaline (Widrow y Hoff, 1960)
- Perceptrón multicapa (Multilayer perceptron) (MLP)
- Backpropagation (Rumelhart, Hinton y Williams, 1986)
- Correlación en cascada (Cascade correlation) (Fahlman y Lebiere, 1990)

(continuación)

- Quickpropagation (Quickprop) (Fahlman, 1988)
- Delta-bar-delta (Jacobs, 1988)
- Resilient Propagation (RPROP) (Riedmiller y Braun, 1993)
- Gradiente conjugado (Battiti, 1992)

- Radial Basis Function (RBF) (Broomhead y Lowe, 1988; Moody y Darken, 1989)
- Orthogonal Least Squares (OLS) (Chen, Cowan y Grant, 1991)

- Cerebellar Articulation Controller (CMAC) (Albus, 1975)
- Sólo clasificación:
- Learning Vector Quantization (LVQ) (Kohonen, 1988)
- Red Neuronal Probabilística (PNN) (Probabilistic Neural Network) (Specht, 1990)

- Sólo regresión:
- General Regression Neural Network (GRNN) (Specht, 1991)

2. Con conexiones feedback

- Bidirectional Associative Memory (BAM) (Kosko, 1992)
- Máquina de Boltzman (Ackley, Hinton y Sejnowski, 1985)
- Series temporales recurrentes
- Backpropagation through time (Werbos, 1990)
- Elman (Elman, 1990)
- Finite Impulse Response (FIR) (Wan, 1990)
- Jordan (Jordan, 1986)
- Real-time recurrent network (Williams y Zipser, 1989)
- Recurrent backpropagation (Pineda, 1989)
- Time Delay NN (TDNN) (Lang, Waibel y Hinton, 1990)

3. Competitivo

- ARTMAP (Carpenter, Grossberg y Reynolds, 1991)
- Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds y Rosen, 1992)
- Gaussian ARTMAP (Williamson, 1995)
- Counterpropagation (Hecht-Nielsen 1987, 1988, 1990)
- Neocognitrón (Fukushima, Miyake e Ito, 1983; Fukushima, 1988)

2. No supervisado

1. Competitivo

- Vector Quantization
- Grossberg (Grossberg, 1976)
- Kohonen (Kohonen, 1984)
- Conscience (Desieno, 1988)

- Mapa Auto-Organizado (Self-Organizing Map) (Kohonen, 1982; 1995)
- Teoría de la Resonancia Adaptativa (Adaptive Resonance Theory, ART)
- ART 1 (Carpenter y Grossberg, 1987a)
- ART 2 (Carpenter y Grossberg, 1987b)
- ART 2-A (Carpenter, Grossberg y Rosen, 1991a)
- ART 3 (Carpenter y Grossberg, 1990)
- Fuzzy ART (Carpenter, Grossberg y Rosen (1991b)

- Differential Competitive Learning (DCL) (Kosko, 1992)

(continuación)

2. Reducción de dimensionalidad
   - Regla de Oja (Oja, 1989)
   - Sanger (Sanger, 1989)
   - Differential hebbian (Kosko, 1992)
3. Autoasociación
   - Autoasociador lineal (Anderson, Silverstein, Ritz y Jones, 1977)
   - Brain-State-in-a-Box (BSB) (Anderson, Silverstein, Ritz y Jones, 1977)
   - Red de Hopfield (1982)

Para discriminar el tipo de aplicación de las RNA que se realiza mayoritariamente, los 2.057 registros se clasificaron en una de 43 áreas o disciplinas de aplicación, siendo las más frecuentes y por este orden: medicina (637 registros), ingeniería (597 registros), biología (362 registros) y psicología (132 registros) (ver gráfico 3). Las áreas de aplicación abarcan prácticamente cualquier disciplina de conocimiento (alimentación, aviación, agricultura, arqueología, documentación, hidrología, medio ambiente, música, tráfico, veterinaria, etc.).

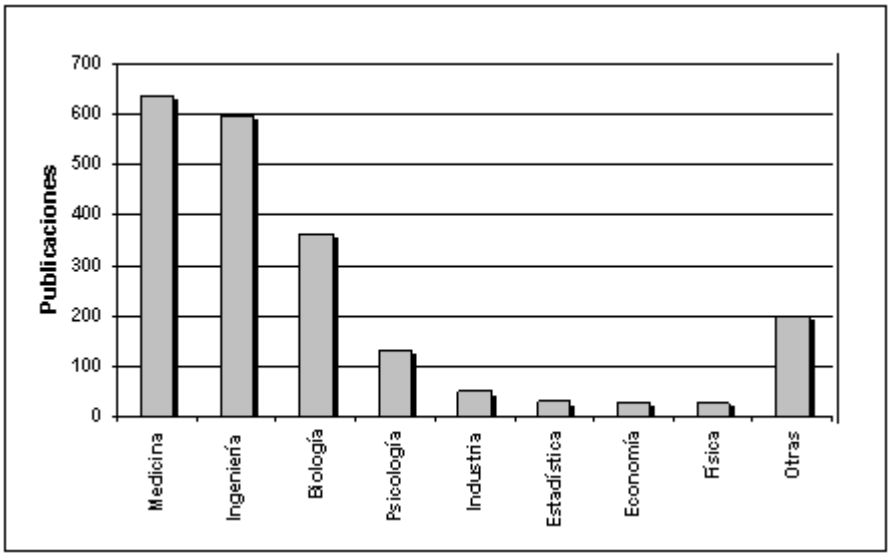


Gráfico 3. Áreas de aplicación más frecuentes (Cajal et al., 2001).

En general, las aplicaciones estudiadas tienen por objeto el reconocimiento de patrones, tanto en la vertiente de clasificación como de estimación de variables continuas. La red neuronal más ampliamente utilizada en las aplicaciones es el MLP asociado al

algoritmo *backpropagation* orientado a la clasificación, que supone aproximadamente el 80% de estos trabajos.

Hemos podido observar en el gráfico que el campo de aplicación mayoritario es la medicina. En este campo, las RNA se han utilizado principalmente en el diagnóstico o discriminación de pacientes con algún tipo de patología a partir de una serie de variables de entrada a la red, susceptibles de poder explicar el estatus del sujeto. En este sentido, el trabajo de Baxt (1991) puede considerarse pionero con la aplicación de una red MLP en la clasificación de pacientes con cardiopatía en función de un conjunto de variables explicativas. Otras áreas de aplicación mayoritarias son la ingeniería con importantes trabajos en el campo de la energía nuclear (Guo y Uhrig, 1992; Bahbah y Girgis, 1999) y la biología con la identificación de cadenas de ADN (Ogura, Agata, Xie, Odaka y Furutani, 1997; Choe, Ersoy y Bina, 2000).

1.2.2. Aplicación de RNA en Psicología.

La aplicación de las RNA en el campo de la Psicología puede considerarse como incipiente en comparación a otros campos de aplicación. En este sentido, realizamos un examen mediante un análisis de contenido sobre el papel que desempeñan las RNA en las diferentes áreas de nuestra disciplina. Para ello, nos centramos en el estudio de los 132 registros que tratan sobre la aplicación de RNA en este ámbito.

Así, observamos que en el área de Evaluación, Personalidad y Tratamiento los autores se interesan principalmente por el diagnóstico de trastornos mentales (Zou et al., 1996). Un ejemplo ilustrativo lo ofrece el trabajo de Pitarque, Ruíz, Fuentes, Martínez y García-Merita (1997), quienes han desarrollado una RNA del tipo MLP con el objeto de clasificar un grupo de sujetos en una de cuatro categorías diagnósticas (depresivo, esquizofrénico, neurótico o mentalmente sano) a partir de las respuestas dadas a un cuestionario elaborado por los autores en base a criterios diagnósticos. El modelo resultante fue capaz de clasificar correctamente el 91.7 % del conjunto de test. Por su parte, el equipo de Buscema (1995) ha desarrollado, de forma pionera, un conjunto de RNA dirigidas a la predicción del consumo de drogas, obteniendo resultados muy satisfactorios. Como veremos más adelante, nuestro equipo ha continuado esta línea de investigación aplicando redes MLP al consumo de éxtasis en la población de jóvenes europeos.

En el área de Metodología los temas prioritarios versan sobre la aplicación de RNA al reconocimiento de patrones (clasificación y predicción) y su comparación con modelos estadísticos clásicos mediante simulación. El equipo de Pitarque (Pitarque, Roy y Ruíz, 1998) ha realizado una comparación entre redes MLP y modelos estadísticos (regresión múltiple, análisis discriminante y regresión logística) en tareas de predicción y clasificación (binaria o no binaria), manipulando los patrones de correlación existentes entre los predictores (o variables de entrada) por un lado, y entre predictores y el criterio (variable de salida) por otro. Los resultados mostraron que en tareas de predicción, las RNA y los modelos de regresión múltiple tienden a rendir por igual. Por el contrario, en tareas de clasificación, en todo tipo de condiciones las RNA rinden mejor que los modelos estadísticos de análisis discriminante y regresión logística. Recientemente, Navarro y Losilla (2000) han realizado una comparación entre RNA del tipo MLP y RBF (*Radial Basis Function* o Funciones de Base Radial) (Broomhead y Lowe, 1988) y métodos de imputación clásicos aplicados a la predicción de datos faltantes. Para ello, se generó un conjunto de matrices en las que se manipuló la naturaleza (discreta, ordinal o cuantitativa) y el grado de correlación de las variables, y el porcentaje de valores faltantes. Los resultados ponen de manifiesto que en la mayoría de situaciones las RNA son la técnica de elección para realizar la imputación de datos faltantes.

Por su parte, el área de Procesos Psicológicos Básicos está centrada en el modelado de procesos psicológicos y psicofísicos. Por ejemplo, MacWhinney (1998) se ha centrado en el desarrollo de modelos de adquisición del lenguaje mediante redes neuronales.

Los temas más recurrentes en el área de Psicología Evolutiva tratan sobre la predicción del rendimiento académico (Hardgrave, Wilson y Walstrom, 1994) y la aplicación de modelos conexionistas en educación. En este sentido, Reason (1998) ha hecho uso de modelos PDP para crear programas de enseñanza de la lectura y para entender mejor por qué se producen dificultades de lectura en niños.

En el área de Psicología Social se trata generalmente de predecir y modelar diferentes conductas sociales como, por ejemplo, el conocido dilema del prisionero (Macy, 1996).

Por último, los autores del área de Psicofisiología se centran en el modelado de procesos psicofisiológicos (Olson y Grossberg, 1998) y en la clasificación de patrones EEG (Grözinger, Kögel y Röschke, 1998). Uno de los autores más prolíficos en esta última línea de investigación es Klöppel (1994).

1.2.3. Aplicación de RNA en el análisis de datos: comparación entre RNA y modelos estadísticos clásicos.

Dado nuestro interés por la aplicación de las RNA en el análisis de datos, de los 549 registros cuya área temática es la comparación entre RNA y otro tipo de modelos (estadísticos, sistemas expertos, etc.), nos centramos en el análisis de los 380 estudios que comparan de forma específica modelos estadísticos y RNA. Siguiendo la sugerencia de Flexer (1995), dividimos este conjunto de trabajos en dos grandes grupos: los que se dedican a hacer comparaciones teóricas (con 32 trabajos) y los que se centran en comparaciones empíricas (con 348 trabajos).

En el primer período de la reemergencia del conexionismo que hemos situado en la segunda mitad de los 80, la idea que se trataba de transmitir consistía en que los modelos neuronales habían surgido como una forma totalmente novedosa de solucionar problemas de clasificación y predicción, sobrepasando siempre en eficacia a las técnicas tachadas de convencionales, como las estadísticas. A lo largo de la década de los 90, una vez reconocido el campo de las RNA ante la comunidad científica, surgieron una serie de trabajos teóricos cuya comparación entre RNA y estadística pone de manifiesto la similitud y, en muchos casos, la identidad entre ambas perspectivas.

Uno de los aspectos que han fomentado la idea errónea acerca de las diferencias entre RNA y estadística versa sobre la terminología utilizada en la literatura de ambos campos. Recordemos que el campo de las RNA surge como una rama de la IA con una fuerte inspiración neurobiológica y su desarrollo ha sido debido a la contribución de investigadores procedentes de una gran variedad de disciplinas. A continuación, se presenta la tabla 3 en la que se pone de manifiesto que las RNA y la estadística utilizan términos diferentes para nombrar el mismo objeto (Sarle, 1994; Vicino, 1998).

De forma análoga, se puede establecer una similitud entre modelos estadísticos y modelos de redes neuronales (ver tabla 4) (Sarle, 1994).

Tabla 3. Equivalencia en la terminología estadística y de redes neuronales.

Terminología estadística	Terminología de redes neuronales
Observación	Patrón
Muestra	Datos de entrenamiento
Muestra de validación	Datos de validación, test
Variables explicativas	Variables de entrada
Variable de respuesta	Variable de salida
Modelo	Arquitectura
Residual	Error
Error aleatorio	Ruido
Estimación	Entrenamiento, aprendizaje
Interpolación	Generalización
Interacción	Conexión funcional
Coeficientes	Pesos de conexión
Constante	Peso umbral
Regresión y análisis discriminante	Aprendizaje supervisado o heteroasociación
Reducción de datos	Aprendizaje no supervisado o autoasociación
Análisis de cluster	Aprendizaje competitivo

Tabla 4. Equivalencia entre modelos estadísticos y modelos de red neuronal.

Modelo estadístico	Modelo de red neuronal
Regresión lineal múltiple	Perceptrón simple con función lineal
Regresión logística	Perceptrón simple con función logística
Función discriminante lineal	Perceptrón simple con función umbral
Regresión no lineal múltiple	Perceptrón multicapa con función lineal en la salida
Función discriminante no lineal	Perceptrón multicapa con función logística en la salida
Análisis de componentes principales	Regla de Oja Perceptrón multicapa autoasociativo
Análisis de clusters	Mapas autoorganizados de Kohonen
K vecinos más cercanos	Learning Vector Quantization (LVQ)
Regresión kernel	Funciones de Base Radial (RBF)

Así, se pone de manifiesto que la mayoría de redes neuronales aplicadas al análisis de datos son similares y, en algunos casos, equivalentes a modelos estadísticos bien conocidos. Vamos a describir las relaciones que se han establecido a nivel teórico entre ambas perspectivas.

Según Sarle (2002), un Perceptrón simple puede ser considerado como un Modelo Lineal Generalizado (MLG) (McCullagh y Nelder, 1989), debido a la equivalencia entre el concepto de función de enlace en un MLG y la función de activación de la neurona de salida en un Perceptrón:

$$Y \equiv f(X, W) \tag{1}$$

donde el valor de la variable de respuesta Y (o variable de salida) se obtiene aplicando una función de enlace (o función de activación) sobre una combinación lineal de coeficientes W (o pesos) y variables explicativas X (o variables de entrada).

La función de enlace en un MLG no suele estar acotada y, en la mayoría de casos, es necesario que sea monótona como las funciones identidad, recíproca y exponencial. Por su parte, la función de activación en un Perceptrón puede estar acotada, como la función sigmoidal logística, o puede no estarlo, como la función identidad; sin embargo, en general todas ellas son monótonas.

El concepto de discrepancia en un MLG y el concepto de función de error en un Perceptrón también son equivalentes (Biganzoli, Boracchi, Mariani y Marubini, 1998). En el caso del Perceptrón la función que en general se intenta minimizar es la suma del error cuadrático:

$$E = \sum_{p=1}^{P} \frac{1}{2} \sum_{k=1}^{M} (d_{pk} - y_{pk})^2 \tag{2}$$

donde P hace referencia al número de patrones, M hace referencia al número de neuronas de salida, d_{pk} es la salida deseada para la neurona de salida k para el patrón p e y_{pk} es la salida obtenida por la red para la neurona de salida k para el patrón p.

Una diferencia importante entre ambos modelos radica en el método de estimación de los coeficientes utilizado para minimizar la función de error. Mientras el Perceptrón normalmente estima los parámetros del modelo mediante el criterio de mínimos cuadrados, es decir, intentando minimizar la función E (White, 1989; Cheng y Titterington, 1994; Ripley, 1994), el MLG ajusta el modelo mediante el método de máxima verosimilitud para una variedad de distribuciones de la clase exponencial (Sarle, 1994). Sin embargo, Bishop (1995), entre otros, ha apuntado que el criterio de mínimos cuadrados asumiendo un error con distribución normal obtiene estimaciones máximo-verosímiles, tal como ocurre en el modelo lineal general. De forma similar, se puede aplicar el método de máxima verosimilitud a un Perceptrón en tareas de clasificación binaria asumiendo un error con distribución de Bernoulli (Hinton, 1989; Spackman, 1992; Van Ooyen y Nienhuis, 1992; Ohno-Machado, 1997; Biganzoli, Boracchi, Mariani y Marubini, 1998). En este caso, la función de error que se intenta minimizar se denomina *cross entropy* (Bishop, 1995) que viene dada por:

$$E = -\sum_{p=1}^{P}\sum_{k=1}^{M}\left[d_{pk}\log y_{pk} + (1-d_{pk})\log(1-y_{pk})\right] \qquad (3)$$

Utilizando esta función de error conseguimos que las salidas puedan ser interpretadas como probabilidades *a posteriori* (Bishop, 1994). Sin embargo, en general la obtención de los parámetros de una red se realiza mediante un criterio de optimización sin tener en cuenta el tipo de distribución de los errores, a diferencia de los MLG (Cheng y Titterington, 1994).

Otra importante diferencia entre RNA y modelos estadísticos consiste en que los parámetros obtenidos por la red neuronal no son susceptibles de una interpretación práctica. No podemos saber inmediatamente cómo los pesos de la red o los valores de activación de las neuronas están relacionados con el conjunto de datos manejados. Así, a diferencia de los modelos estadísticos clásicos, no parece tan evidente conocer en una red el efecto que tiene cada variable explicativa sobre la/s variable/s de respuesta. Por tanto, es importante tener en cuenta que las similitudes que se puedan establecer entre RNA y modelos estadísticos siempre harán referencia al aspecto predictivo pero no al aspecto explicativo. Como veremos más adelante, la problemática acerca del análisis del

efecto de las variables de entrada en una red neuronal constituye una línea de investigación de interés para nuestro equipo.

Estableciendo analogías entre RNA y modelos concretos pertenecientes a MLG, un Perceptrón simple con función de activación lineal en la neurona de salida y utilizando la suma del error cuadrático equivale a un modelo de regresión lineal (Liestol, Andersen y Andersen, 1994; Michie, Spiegelhalter y Taylor, 1994; Sarle, 1994; Kemp, McAulay y Palcic, 1997) (ver figura 5).

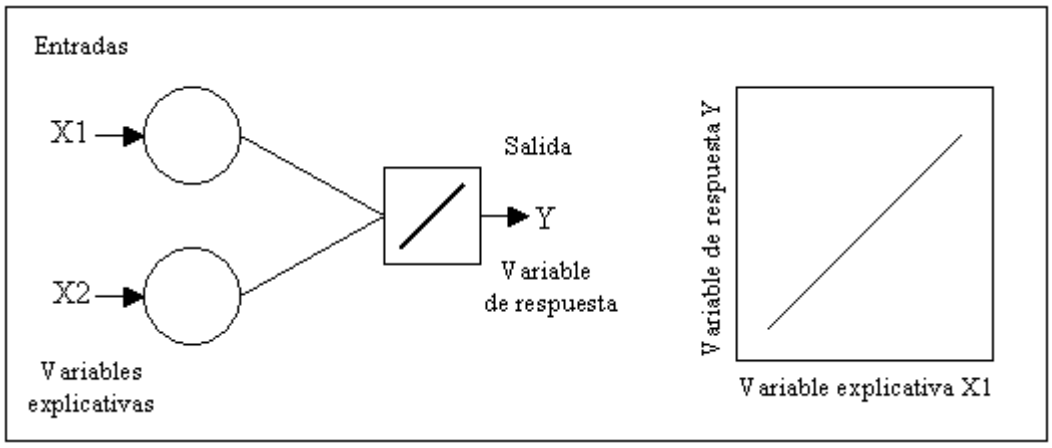


Figura 5. Perceptrón simple con función lineal = Modelo de regresión lineal.

La representación matemática de este tipo de red resulta muy familiar desde el punto de vista estadístico y viene dada por:

$$y_{pk} = \theta_k + \sum_{i=1}^{N} w_{ik} \cdot x_{pi} \qquad (4)$$

donde θ_k es el umbral de la neurona de salida k que actúa de forma similar a la constante del modelo de regresión, N hace referencia al número de neuronas de entrada, w_{ik} es el peso entre la neurona de entrada i y la neurona de salida k y x_{pi} es el valor de la neurona de entrada i para el patrón p.

Si hay más de una neurona de salida, la red se convierte en un modelo de regresión multivariado.

Por su parte, un Perceptrón simple con función de activación logística en la neurona de salida es similar a un modelo de regresión logística (Sarle, 1994) (ver figura 6).

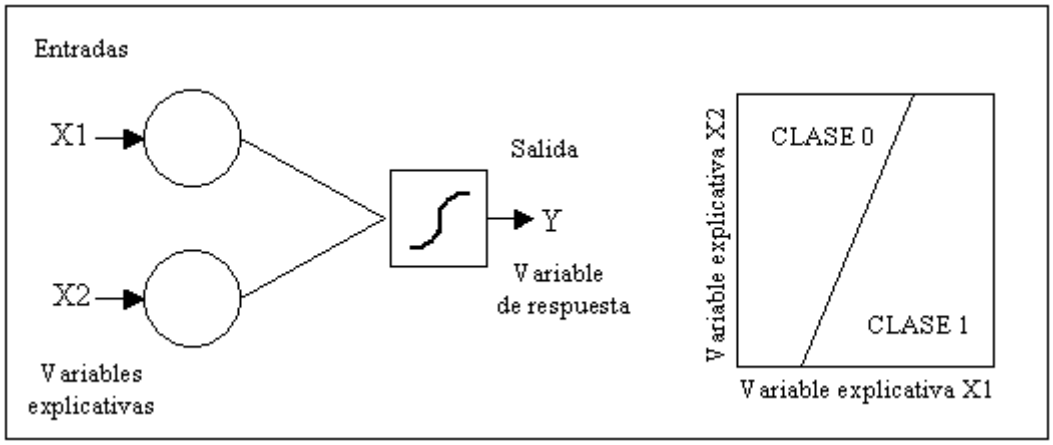


Figura 6. Perceptrón simple con función logística = Modelo de regresión logística.

La representación matemática de este tipo de red viene dada por:

$$y_{pk} = \frac{1}{1 + \exp\left(-\theta_k + \sum_{i=1}^{N} w_{ik} \cdot x_{pi} \right)} \tag{5}$$

Si la función de error que intentamos minimizar es la *cross entropy* expresada en la ecuación (3), bajo ciertas condiciones, el algoritmo de obtención de los pesos equivale al método de estimación de máxima verosimilitud y la salida de la red se puede interpretar como una probabilidad *a posteriori*.

Un Perceptrón simple con función de activación umbral en la neurona de salida es similar a la Función Discriminante Lineal de Fisher (Kemp, McAulay y Palcic, 1997) (ver figura 7).

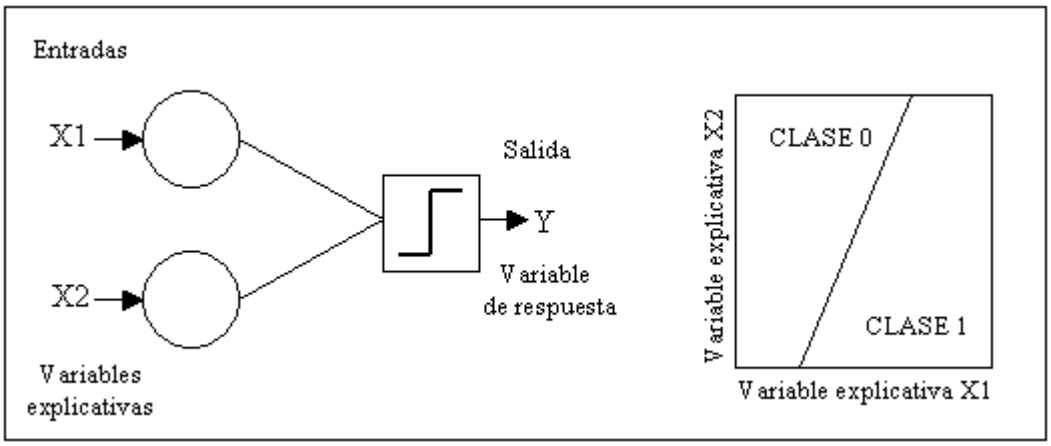


Figura 7. Perceptrón simple con función umbral = Análisis discriminante lineal.

La equivalencia se da cuando las observaciones pertenecientes a las dos categorías se distribuyen según la distribución normal con covariancias idénticas y probabilidades *a priori* iguales.

Si hay más de una neurona de salida, la red se convierte en una función discriminante múltiple. En estos casos, normalmente se utiliza una función logística múltiple denominada función de activación *softmax* o exponencial normalizada (*normalized exponential*), en lugar de una función umbral múltiple. La función *softmax* permite que las múltiples salidas de la red puedan ser interpretadas como probabilidades a posteriori, de forma que la suma de todas ellas es igual a 1. La función *softmax* se expresa como:

$$y_{pk} = \frac{\exp(net_{pk})}{\sum_{k=1}^{M} \exp(net_{pk})} \qquad (6)$$

donde net_{pk} es la entrada neta que recibe la neurona de salida k.

La función (6) supone una generalización de la función logística aplicada a una única neurona de salida que representa dos categorías.

Una red MLP compuesta por tres capas cuya capa oculta de neuronas utiliza una función de activación no lineal –en general, la función logística--, puede ser vista como una generalización no lineal de los MLG (Biganzoli, Boracchi, Mariani y Marubini, 1998).

La principal virtud de una red MLP que permite explicar su amplio uso en el campo del análisis de datos es que se trata de un aproximador universal de funciones. La base matemática de esta afirmación se debe a Kolmogorov (1957), quien constató que una función continua de diferentes variables puede ser representada por la concatenación de varias funciones continuas de una misma variable. Esto significa que un Perceptrón conteniendo al menos una capa oculta con suficientes unidades no lineales, tiene la capacidad de aprender virtualmente cualquier tipo de relación siempre que pueda ser aproximada en términos de una función continua (Cybenko, 1989; Funahashi, 1989; Hornik, Stinchcombe y White, 1989). También se ha demostrado que utilizando más de una capa oculta, la red puede aproximar relaciones que impliquen funciones discontinuas (Rzempoluck, 1998). Si no se utilizan funciones de activación no lineales

en la/s capa/s oculta/s, la red queda limitada a actuar como un discriminador/aproximador lineal.

Otra propiedad importante de las redes MLP es que son capaces de manejar tareas de elevada dimensionalidad mediante la utilización de arquitecturas relativamente sencillas. Esta propiedad está relacionada con el hecho de que no es necesario introducir explícitamente en el modelo las interacciones entre las variables explicativas, ya que las posibles interacciones son aprendidas por la red neuronal de forma automática en el proceso de entrenamiento.

Por último, hemos comentado que las RNA estiman los pesos en base a algún criterio de optimización sin tener en cuenta supuestos como el tipo de distribución o la dependencia funcional entre las variables. Por este motivo, las RNA han sido consideradas por muchos autores como modelos no paramétricos (Smith, 1993). Sin embargo, autores de reconocido prestigio como Bishop (1995) sostienen que las RNA y los modelos estadísticos asumen exactamente los mismos supuestos en cuanto al tipo de distribución; lo que sucede es que los estadísticos estudian las consecuencias del incumplimiento de tales supuestos, mientras que los investigadores de RNA simplemente las ignoran. En este sentido, hemos visto el paralelismo que se establece entre los criterios de minimización utilizados por las RNA y el método de máxima-verosimilitud, bajo el cumplimiento de ciertos supuestos. Otros autores como Masters (1993) son más flexibles y sostienen que supuestos como normalidad, homogeneidad de variancias y aditividad en las variables de entrada son características recomendables para una red neuronal aunque no son estrictamente necesarias como sucede en los modelos estadísticos.

Este conjunto de propiedades convierten las redes MLP en herramientas de propósito general, flexibles y no lineales.

Dependiendo del tipo de función de activación utilizado en la capa de salida, el MLP se puede orientar a la predicción o a la clasificación. Así, en caso de utilizar la función identidad en la capa de salida, estaríamos ante un modelo de regresión no lineal (Cheng y Titterington, 1994; Ripley, 1994; Flexer, 1995) (ver figura 8).

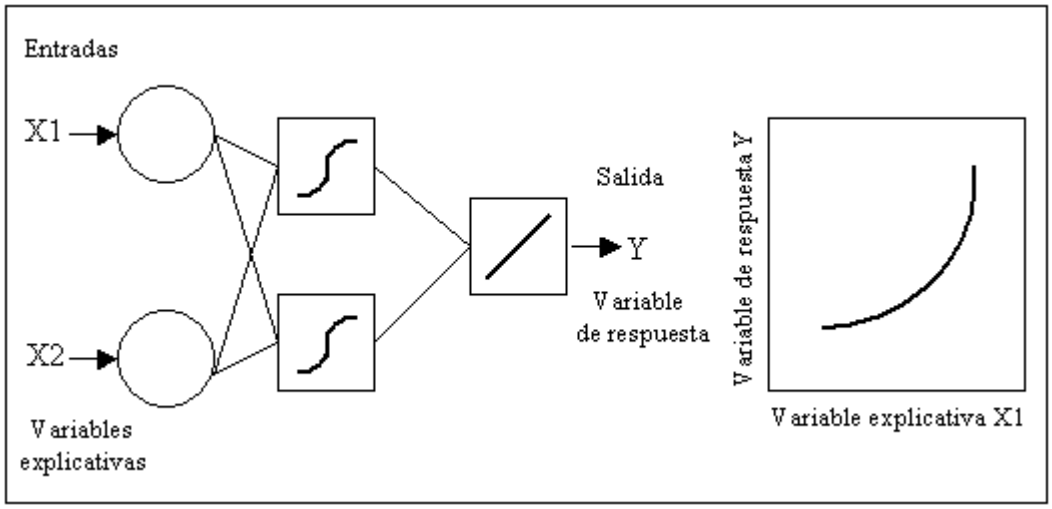


Figura 8. Perceptrón multicapa con función lineal en la salida = Modelo de regresión no lineal.

La representación matemática de este tipo de arquitectura viene dada por:

$$y_{pk} = f_M\left(\theta_k + \sum_{j=1}^{L} v_{jk} \cdot f_L\left(\theta_j + \sum_{i=1}^{N} w_{ij} \cdot x_{pi}\right)\right) \qquad (7)$$

donde f_M y f_L son las funciones de activación de las M neuronas de salida y las L neuronas ocultas, respectivamente; θ_j es el umbral de la neurona oculta j, w_{ij} es el peso entre la neurona de entrada i y la neurona oculta j, y v_{jk} es el peso entre la neurona oculta j y la neurona de salida k.

Una red MLP con funciones de activación logísticas en las salidas puede ser utilizada como una Función Discriminante no lineal (Biganzoli, Boracchi, Mariani y Marubini, 1998) (ver figura 9).

Como se puede observar en la figura, cada neurona oculta corresponde a un límite no lineal entre la clase 0 y la clase 1. Así, la utilización de un número considerable de neuronas ocultas permite obtener regiones de decisión arbitrariamente complejas.

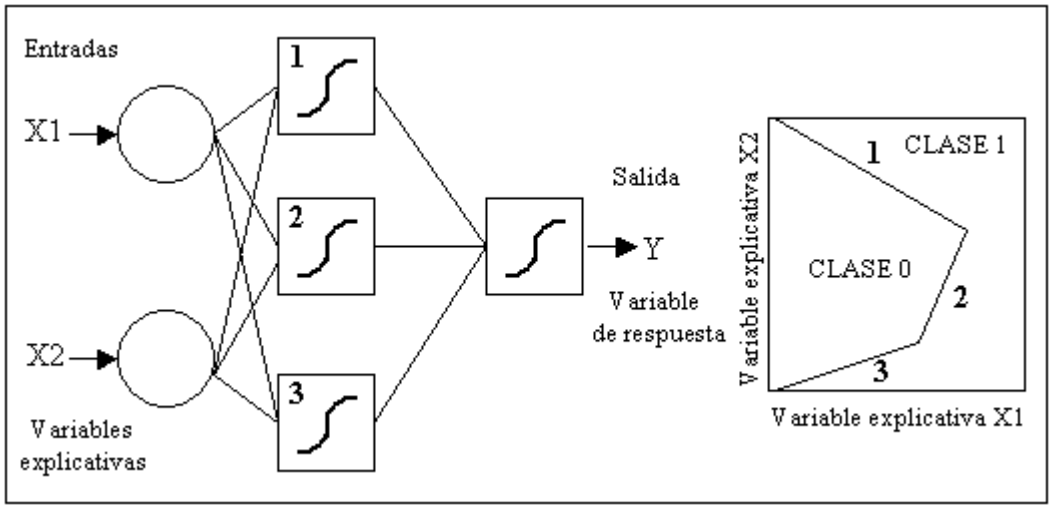


Figura 9. Perceptrón multicapa con función logística = Función discriminante no lineal.

Como se ha comentado, el método más utilizado para la estimación de los pesos de esta arquitectura de red es el *backpropagation* o técnica de gradiente decreciente basado en el criterio de mínimos cuadrados. Esta técnica ya fue utilizada en el campo de la estadística antes del desarrollo de las RNA (Smith, 1993). Se puede encontrar una descripción de este algoritmo y de algunas de las variantes más conocidas en nuestro artículo *Tutorial sobre redes neuronales artificiales: el Perceptrón multicapa* (Palmer, Montaño y Jiménez, 2001) (ver anexo 1, pág. 239). Sin embargo, el algoritmo *backpropagation* padece dos defectos graves (Masters, 1993). Por un lado, el gradiente es un indicador extremadamente local de la función de cambio óptima. De forma que para zonas próximas entre sí de la superficie del error, el gradiente puede tomar direcciones opuestas. Estas fluctuaciones provocan que el tiempo de búsqueda del mínimo del error sea considerablemente largo, en lugar de tomar una ruta más directa. Por otro lado, no se sabe *a priori* cuál es el tamaño del cambio de los pesos más adecuado para una tarea dada. Este tamaño o magnitud está determinado por los valores de la tasa de aprendizaje y el factor momento. Un ritmo de aprendizaje demasiado pequeño ocasiona una disminución importante en la velocidad de convergencia y un aumento en la probabilidad de acabar atrapado en un mínimo local. En cambio, un ritmo de aprendizaje demasiado grande conduce a inestabilidades en la función de error con el peligro de poder pasar por encima del mínimo global. Por tanto, el valor de estos parámetros de aprendizaje se deben determinar mediante ensayo y error.

En la actualidad se han propuesto, desde los campos de las RNA y la estadística, diversos métodos alternativos al *backpropagation* dirigidos a estimar los pesos de la red

de una forma mucho más rápida y eficaz. Muchos de estos métodos alternativos son extensiones de la propia técnica de gradiente decreciente como la regla *delta-bar-delta* basada en la utilización de tasas de aprendizaje adaptativas aplicadas al valor del gradiente (Jacobs, 1988), el algoritmo RPROP (*Resilient propagation*) (Riedmiller y Braun, 1993) basado en un método de aprendizaje adaptativo parecido a la regla *delta-bar-delta*, donde los pesos se modifican en función del signo del gradiente, no en función de su magnitud y, finalmente, el algoritmo *Quickprop* (Fahlman, 1988) basado en modificar los pesos en función del valor del gradiente obtenido en la iteración actual y del gradiente obtenido en la iteración anterior.

Por otra parte, existe un conjunto de algoritmos de optimización no lineal derivados del campo del análisis numérico y la estadística (Gill, Murray y Wright, 1981; Fletcher, 1987; Bertsekas, 1995; Bertsekas y Tsitsiklis, 1996) que se caracterizan por hacer uso no sólo de la información proporcionada por la derivada de primer orden del error con respecto a los pesos, como es el caso del *backpropagation* y sus extensiones, sino también de la información proporcionada por la derivada de segundo orden (Rojas, 1996). Mientras la derivada de primer orden informa de la pendiente de la superficie del error, la derivada de segundo orden informa de la curvatura de dicha superficie. Así, si la primera derivada representa la velocidad de decremento, la segunda derivada representa la deceleración del error. Este aspecto se ilustra en la figura 10.

Se muestran dos superficies de error (línea gruesa y línea punteada) alineadas de forma que se tocan en un punto en el que ambas derivadas de primer orden son iguales, indicado por la línea recta tangente a las dos curvas. La tasa de decremento de las dos superficies es el mismo. Sin embargo, sus curvaturas no son iguales: la superficie de trazo grueso tiene una curvatura mayor y, por tanto, está más cerca del mínimo en relación a la superficie de trazo punteado. La información adicional proporcionada por la derivada de segundo orden puede servir para acelerar o decelerar el descenso por la superficie dependiendo de la distancia a la cual se encuentre respecto al mínimo.

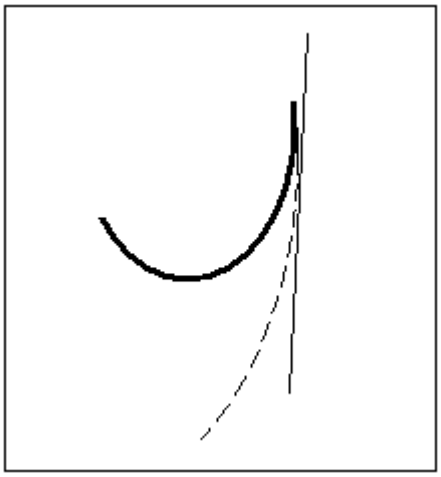


Figura 10. Dos superficies de error.

El conjunto de algoritmos que hacen uso no sólo de la información de la pendiente, sino también de la curvatura de la superficie se denominan genéricamente métodos de segundo orden. Dentro de este grupo, uno de los métodos más conocidos en el campo de las RNA es el algoritmo de gradientes conjugados (Battiti, 1992), el cual se basa en dividir la derivada de primer orden por la derivada de segundo orden para determinar el incremento de cada peso:

$$\Delta w_{ij} = -\frac{\partial E / \partial w_{ij}}{\partial^2 E / \partial w_{ij}^2} \tag{8}$$

Este incremento representa la distancia necesaria para que la deceleración determine una velocidad igual a 0, que es el punto en el que se alcanza un mínimo en la función de error. Esta estrategia permite acelerar el proceso de aprendizaje de forma considerable con respecto a los métodos anteriores, alcanzando la convergencia de los parámetros de una forma más eficaz y directa. Otros métodos de segundo orden son los algoritmos de Newton, cuasi-Newton, Gauss-Newton y Newton-Raphson (Bishop, 1995). La limitación de estos métodos es que su uso requiere un alto nivel de experiencia (Smith, 1993). Por otra parte, algunos son computacionalmente intensivos y, como consecuencia, se deberían utilizar en aquellos casos en que el número de pesos a estimar no es muy grande (Sarle, 2002).

Existen otros modelos de RNA, a partir de los cuales también se puede establecer una clara analogía con modelos estadísticos clásicos conocidos. Este es el caso de las redes entrenadas mediante la regla de Oja (1982, 1989), las cuales permiten realizar Análisis de Componentes Principales (PCA). La regla de Oja supone una modificación de la regla no supervisada de Hebb. La arquitectura de este tipo de red está compuesta por una capa de entrada con N neuronas y una capa de salida con M neuronas lineales (ver figura 11).

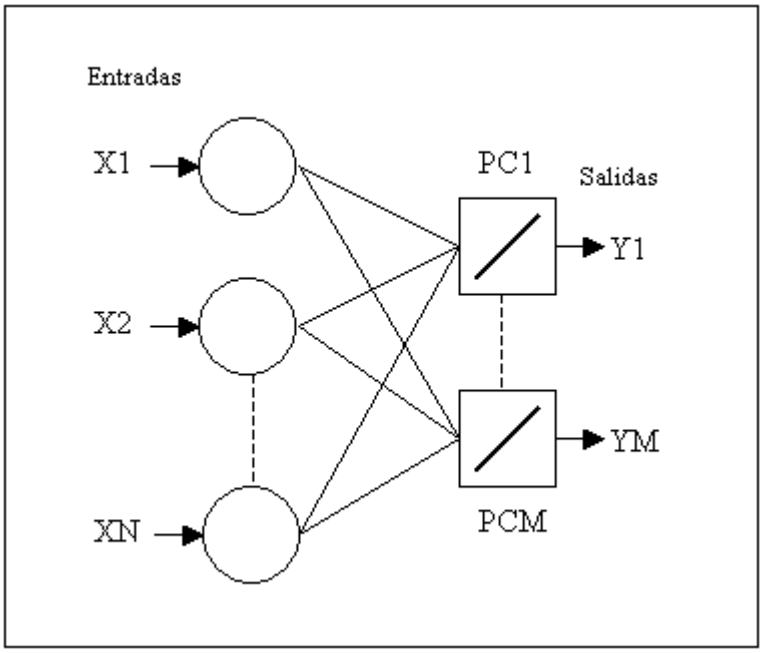


Figura 11. Regla de Oja = Análisis de componentes principales.

Las neuronas de salida representan los M primeros componentes principales y el vector de pesos asociado a cada una de ellas representa el vector propio. Los vectores propios resultantes tienen una longitud igual a la unidad sin necesidad de realizar una normalización y son ortogonales entre sí. La modificación de los pesos se realiza de forma iterativa mediante la siguiente expresión:

$$\Delta w_{ik} = \eta\, y_{pk}\left(x_{pi} - y_{pk} w_{ik}\right) \tag{9}$$

donde η es la tasa de aprendizaje e $y_{pk} = \sum_{i=1}^{N} w_{ik} x_{pi}$ es la salida de la neurona de salida k para el patrón p. El valor propio λ asociado a un vector propio W se obtiene mediante (Hertz, Krogh y Palmer, 1991):

$$\lambda = W^T \cdot C \cdot W \tag{10}$$

donde C es la matriz de correlaciones.

La red *backpropagation* autosupervisada o MLP autoasociativo es otro modelo de red que también ha sido aplicado al PCA y a la reducción de la dimensionalidad. Esta red fue utilizada inicialmente por Cottrell, Munro y Zipser (1989) para la compresión de imágenes y ha sido aplicada en el campo de la ingeniería (Garrido, Gaitan, Serra y Calbet, 1995). A pesar de no ser muy conocida en el ámbito del análisis de datos, creemos que esta red puede ser de gran utilidad ya que, como vamos a ver, puede superar algunas de las limitaciones que presenta el PCA clásico.

Como se puede observar en la figura 12, la red *backpropagation* autosupervisada consiste en una arquitectura MLP de tres capas en donde la salida deseada es igual a la información de entrada. Se trata, por tanto, de una red autoasociativa ya que la red es entrenada para reproducir la información de entrada.

En este tipo de red las unidades ocultas se denominan neuronas “cuello de botella” debido a que los vectores de entrada de N dimensiones deben pasar a través de una capa de menor dimensión antes de ser reproducidos en la salida de la red.

El interés de este modelo no reside en la salida que proporciona, sino en la representación interna que se genera en la capa oculta. Más concretamente, la red realiza una proyección ortogonal del espacio multidimensional de entrada N sobre un subespacio determinado por los L primeros componentes principales, donde L es el número de unidades ocultas (Hertz, Krogh y Palmer, 1991). Baldi y Hornik (1989) demostraron que la correcta proyección del espacio de entrada se obtiene cuando la red alcanza el mínimo global de la función de error cuadrático mediante la utilización del algoritmo *backpropagation*. Así, cada patrón de entrada es proyectado en un espacio de L dimensiones (en general, una, dos o tres dimensiones para poder ser fácilmente visualizado) a partir de los valores de activación que proporcionan las L neuronas

ocultas. Los vectores de pesos que conectan la capa de entrada con las neuronas ocultas representan los primeros vectores propios de la matriz de correlaciones de las variables de entrada. A este mismo resultado se llega utilizando la regla de Oja (1982, 1989) descrita anteriormente (Baldi y Hornik, 1989). Según Bishop (1995), la similitud con el procedimiento clásico no es sorprendente debido a que tanto el PCA como esta red autoasociativa usan una reducción de la dimensionalidad lineal e intentan minimizar la misma función de error cuadrático.

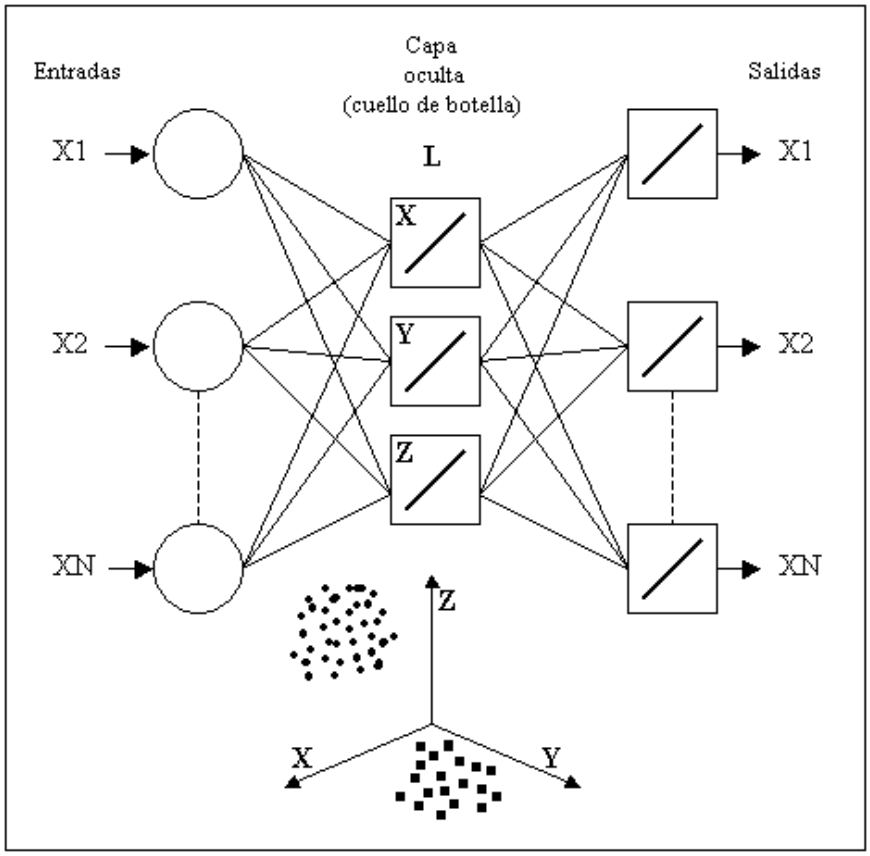


Figura 12. Perceptrón multicapa autoasociativo = Análisis de Componentes Principales.

Se podría pensar que las limitaciones de la reducción de la dimensionalidad lineal pueden ser superadas mediante el uso de funciones de activación no lineales en la capa oculta al igual que en el caso de tareas de clasificación o predicción descritas. Sin embargo, Bourland y Kamp (1988) han mostrado que la introducción de funciones no lineales no confiere ninguna ventaja en el resultado final.

Consideremos añadir dos capas de neuronas intermedias al modelo autoasociativo anterior como se muestra en la figura 13.

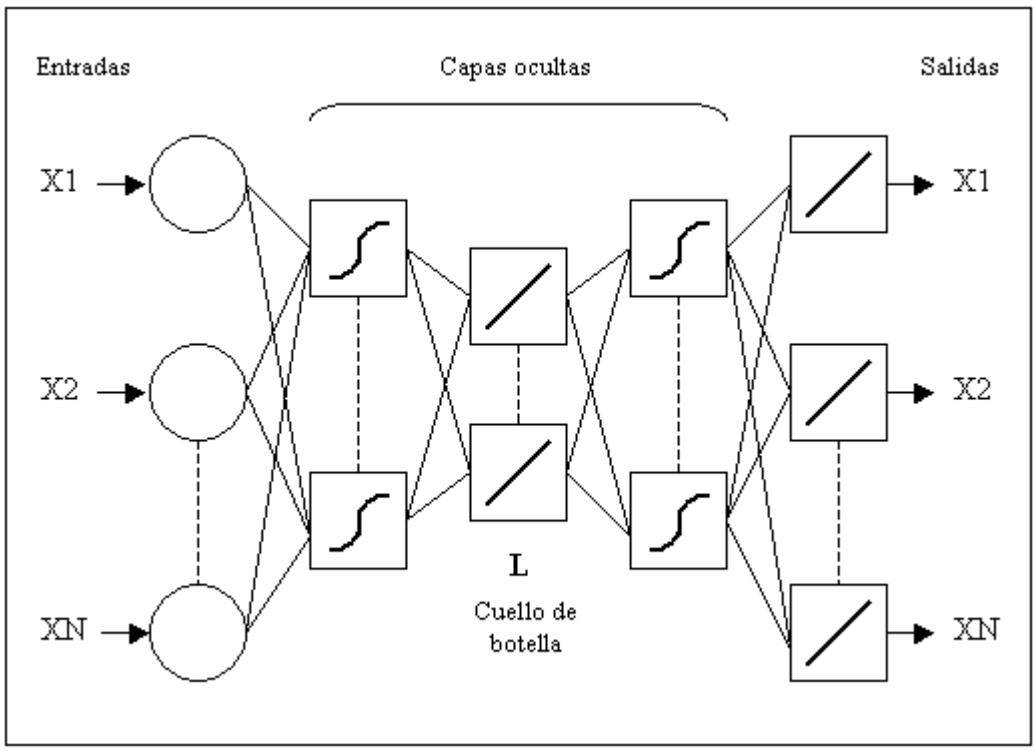


Figura 13. Perceptrón multicapa autoasociativo con tres capas ocultas = Análisis de Componentes Principales no lineal.

Observese que, de nuevo, la capa de salida y la capa central (o "cuello de botella") utilizan una función lineal. En cambio, las capas adicionales (la segunda y la cuarta capa) utilizan una función no lineal de tipo sigmoidal. Kramer (1991) demostró que esta configuración permite realizar con eficiencia una proyección no lineal del espacio multidimensional de entrada N sobre un subespacio determinado por las L neuronas de la capa central. Por tanto, se puede decir que esta red supone una generalización no lineal del PCA clásico, con la ventaja de no estar limitada a transformaciones lineales. Debido a la complejidad de la arquitectura, es conveniente estimar los parámetros mediante alguna técnica de optimización no lineal de las comentadas anteriormente, en lugar del clásico *backpropagation* (Fotheringhame y Baddeley, 1997).

Por otra parte, hay algunos modelos de red, como los mapas autoorganizados o SOM (*Self-Organizing Maps*) (Kohonen, 1982) y el *Learning Vector Quantization* (LVQ) (Kohonen, 1988), que a pesar de no tener un equivalente estadístico preciso, han sido ampliamente utilizados en el análisis de datos.

Una descripción detallada de la arquitectura, aprendizaje y funcionamiento de los mapas autoorganizados se puede encontrar en nuestro artículo *Tutorial sobre redes neuronales*

artificiales: los mapas autoorganizados de Kohonen (Palmer, Montaño y Jiménez, 2002) (ver anexo 1, pág. 271). Desde un punto de vista estadístico, este tipo de redes permiten realizar análisis de clusters proyectando un espacio multidimensional de entrada sobre otro de dimensión mucho menor (en general un mapa bidimensional) de salida (ver figura 14).

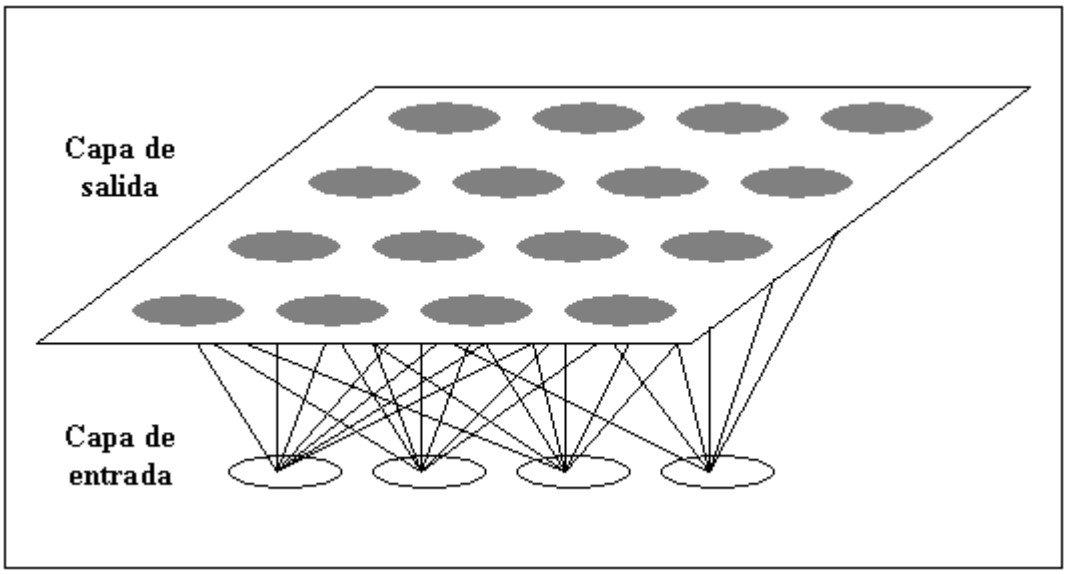


Figura 14. Arquitectura de un mapa autoorganizado.

Para ello, se vale de un aprendizaje no supervisado de tipo competitivo en el que cada patrón de entrada se asocia a la neurona de salida cuyo vector de referencia o vector prototipo de pesos es el más parecido. En general, el criterio de similitud más utilizado es la distancia euclídea que viene dado por la siguiente expresión:

$$\min\left\|X_p - W_j\right\| = \min \sum_{i=1}^{N} (x_{pi} - w_{ij})^2 \qquad (11)$$

Como resultado tenemos que los patrones de entrada se agrupan en diversas zonas del mapa bidimensional de salida, de forma que los patrones situados próximos entre sí son los patrones que presentan características en común respecto a las variables de entrada.

Por su parte, el *Learning Vector Quantization* (LVQ) (Kohonen, 1988) actúa como un clasificador de patrones mediante la aplicación de una versión supervisada de la regla de aprendizaje utilizada en los mapas autoorganizados. Así, de forma similar a estos últimos, cada patrón de entrada es clasificado en la categoría representada por la

neurona de salida cuyo vector prototipo de pesos es el más parecido a ese patrón en base a la distancia euclídea. Por tanto, el LVQ utiliza un algoritmo muy parecido a la técnica estadística K vecinos más cercanos. La diferencia entre ambos modelos radica en que mientras la técnica K vecinos más cercanos clasifica un determinado patrón en función de su similitud con otros patrones de entrenamiento, el LVQ clasifica el patrón en función de su similitud con vectores prototipo asociados a neuronas de salida (Michie, Spiegelhalter y Taylor, 1994).

Finalmente, también se pueden establecer semejanzas entre determinados modelos estadísticos y las redes de función de base radial (*Radial Basis Function*: RBF) (Broomhead y Lowe, 1988) que constituyen la red más usada en problemas de predicción y clasificación, tras la red MLP. Como se puede observar en la figura 15, las RBF están compuestas de tres capas al igual que la red MLP.

La particularidad de las RBF reside en que las neuronas ocultas operan en base a la distancia euclídea que separa el vector de entrada X_p respecto al vector de pesos W_j que cada una almacena (denominado centroide), cantidad a la que aplican una función radial con forma gaussiana, de forma similar a las funciones kernel en el modelo de regresión kernel (Bishop, 1995).

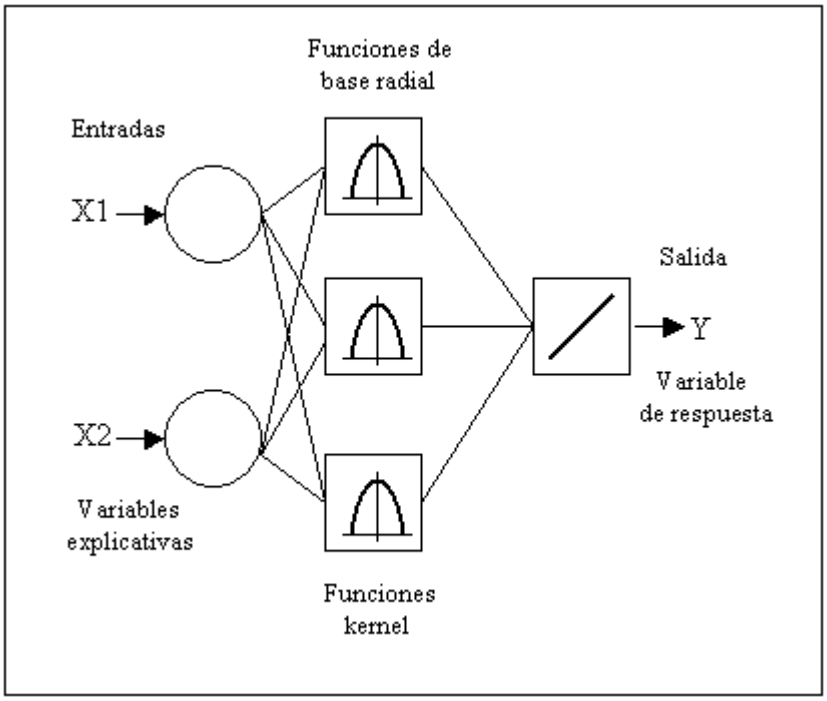


Figura 15. Red de función de base radial = Modelo de regresión kernel.

La representación matemática de la función radial que aplican las neuronas ocultas sobre el vector de entrada viene definida por:

$$b_{pj} = \exp\left[\frac{-\sum_{i=1}^{N}(x_{pi} - w_{ij})^2}{2\sigma^2}\right] \qquad (12)$$

Si el vector de entrada X_p coincide con el centroide W_j de la neurona j, ésta responde con máxima salida (la unidad). Es decir, cuando el vector de entrada está situado en una región próxima al centroide de una neurona, ésta se activa, indicando que reconoce el patrón de entrada; si el patrón de entrada es muy diferente del centroide, la respuesta tenderá a cero.

El parámetro de normalización σ (o factor de escala) mide la anchura de la gaussiana, y equivaldría al radio de influencia de la neurona en el espacio de las entradas; a mayor σ la región que la neurona domina en torno al centroide es más amplia.

La salida de las neuronas de salida se obtiene como una combinación lineal de los valores de activación de las neuronas ocultas ponderados por los pesos que conectan ambas capas de igual forma que la expresión matemática (4) asociada al Perceptrón simple:

$$y_{pk} = \theta_k + \sum_{j=1}^{L} v_{jk} \cdot b_{pj} \qquad (13)$$

Como la red MLP, las RBF permiten realizar con relativa facilidad modelados de sistemas no lineales arbitrarios y también constituyen aproximadores universales de funciones (Hartman, Keeler y Kowalski, 1990), con la particularidad de que el tiempo requerido para su entrenamiento suele ser mucho más reducido. Esto es debido en gran medida a que las redes RBF dividen el aprendizaje en dos fases. En una primera fase, los vectores de pesos o centroides asociados a las neuronas ocultas se pueden obtener mediante un aprendizaje no supervisado a través de un método estadístico como el algoritmo k-medias o a través de un método propio del campo de las RNA como el algoritmo de Kohonen utilizado en los mapas autoorganizados. En una segunda fase, los

pesos de conexión entre las neuronas ocultas y las de salida se obtienen mediante un aprendizaje supervisado a través de la regla delta de Widrow-Hoff (1960). Considerando las redes RBF como modelos de regresión no lineales, los pesos también podrían ser estimados por un método convencional como mínimos cuadrados no lineales o máxima-verosimilitud.

A la familia de las redes RBF pertenecen, entre otras, la Red Neuronal Probabilística (PNN, *Probabilistic Neural Network*) (Specht, 1990), la *General Regression Neural Network* (GRNN) (Specht, 1991) y la *Counterpropagation* (Hecht-Nielsen, 1987, 1988, 1990).

La conclusión a la que podemos llegar acerca de la comparación teórica entre RNA y modelos estadísticos es que no se trata de metodologías contrapuestas (White, 1989). Se ha puesto de manifiesto que existe un solapamiento entre ambos campos. Las RNA incluyen diversos modelos, como el MLP, que son de gran utilidad en las aplicaciones de análisis de datos y estadística. La metodología estadística es directamente aplicable a las RNA de diversas formas, incluyendo criterios de estimación y algoritmos de optimización.

Respecto a los trabajos centrados en la comparación a nivel empírico entre RNA y modelos estadísticos, se realizó una clasificación en función del objetivo que perseguían:

- Clasificación: asignación de la categoría de pertenencia de un determinado patrón y agrupamiento de patrones en función de las características comunes observadas entre los mismos.
- Predicción: estimación de variables cuantitativas.
- Exploración/reducción: identificación de factores latentes y reducción de espacios de alta dimensión.

Los resultados reflejan que los trabajos cuyo objetivo es la clasificación representan el 71% de este tipo de estudios comparativos. Entre este conjunto, los trabajos más sobresalientes que constituyen puntos de referencia son los de Thrun, Mitchell y Cheng (1991), Michie, Spiegelhalter y Taylor (1994), Balakrishnan, Cooper, Jacob y Lewis (1994), Waller, Kaiser, Illian y Manry (1998) y Lim, Loh y Shih (1999) y que a continuación pasamos a comentar.

Thrun, Mitchell y Cheng (1991) dirigieron un extenso estudio comparativo de diferentes algoritmos –entre otros, la red MLP, el análisis discriminante y el modelo de regresión logística--, en la clasificación de 432 registros en una de dos categorías a partir de seis atributos. Aunque algunos algoritmos fueron ligeramente superiores, en las conclusiones del trabajo no se destaca especialmente ninguno de ellos.

El trabajo de Michie, Spiegelhalter y Taylor (1994) es considerado como el estudio comparativo más completo entre RNA y modelos estadísticos orientado a la clasificación (Sarle, 2002). En este estudio, se compararon 23 métodos de clasificación pertenecientes a tres perspectivas de investigación diferentes:

- Redes Neuronales Artificiales: Cabe destacar la red MLP, la red RBF, la red SOM y la red LVQ.
- Modelos Estadísticos: Cabe destacar la función discriminante lineal y cuadrática, el método K vecinos más cercanos y el modelo de regresión kernel.
- Algoritmos de Inducción de Reglas: Cabe destacar los algoritmos NewID, AC2, Cal5, CN2, C4.5, CART e IndCART.

Este conjunto de modelos fueron aplicados sobre un total de 22 matrices de datos diferentes. Aunque los resultados de este vasto estudio indican que no existe un método claramente superior en todos los conjuntos de datos analizados, las RNA siempre aparecen dentro del grupo de mejores predictores en función del porcentaje de clasificaciones correctas. Finalmente, los autores apuntan algunos inconvenientes relacionados con las RNA como el excesivo tiempo que tardan algunas redes en obtener los pesos, las exigencias que imponen al usuario –tiene que determinar el número de neuronas ocultas, el valor de los parámetros, la finalización del entrenamiento, etc.--, y la dificultad en comprender la representación interna aprendida por la red.

Balakrishnan, Cooper, Jacob y Lewis (1994) realizaron una comparación entre redes SOM y el método de clusters K-medias en base a un conjunto de datos simulados donde se manipulaba el número de atributos o entradas, el número de clusters y la cantidad de error presente en los datos. Los resultados ponen de manifiesto que, en general, el método K-medias proporciona menos clasificaciones incorrectas que las redes SOM. Por otra parte, estas últimas se ven afectadas negativamente a medida que aumenta el número de clusters, a diferencia del método K-medias. Por su parte, Waller, Kaiser, Illian y Manry (1998) realizaron una comparación entre redes SOM y tres métodos de

clusters jerárquicos mediante varias matrices de datos simuladas. En contraposición a los resultados de Balakrishnan, Cooper, Jacob y Lewis (1994), se pone de manifiesto que las redes SOM tienen un rendimiento similar y, en algunos casos, superior a los métodos de clusters.

Recientemente, Lim, Loh y Shih (1999) han realizado un trabajo similar al de Michie, Spiegelhalter y Taylor (1994), donde se comparan 22 algoritmos de inducción de reglas, nueve modelos estadísticos y dos modelos de red neuronal (RBF y LVQ) sobre un conjunto de 32 matrices de datos. De igual forma, no se observa ningún método que sea sistemáticamente superior a los demás, mostrando en la mayoría de casos rendimientos similares.

Esta serie de resultados poco concluyentes y, en ocasiones, contradictorios son representativos del conjunto de estudios comparativos entre RNA y modelos estadísticos realizados con fines de clasificación. Como posibles causas de estos resultados, hemos podido observar, en primer lugar, que en raras ocasiones se realiza un análisis de la calidad de los datos y normalmente no se comprueba el cumplimiento de los supuestos de los modelos estadísticos. En segundo lugar, con frecuencia los autores han desarrollado su propio algoritmo y son expertos en un tipo de metodología, dando como resultado un sesgo natural en detrimento de otras metodologías de las que no son tan expertos. Por último, los criterios de comparación entre métodos suelen ser poco rigurosos y con sesgos a favor de una determinada metodología.

Respecto a los trabajos sobre predicción, éstos representan el 27% del total de estudios comparativos. En este sentido, se han realizado trabajos sobre la predicción de variables de respuesta de naturaleza continua (Pitarque, Roy y Ruíz, 1998), el análisis de series temporales (Tsui, 1996) y el análisis de supervivencia (Ohno-Machado, 1997).

Por último, los modelos generalmente comparados en los estudios de exploración/reducción son el PCA clásico y los modelos de red entrenados con la regla de Oja (1982), como en el trabajo de Fiori (2000). Estos trabajos representan únicamente el 2% del total.

De los resultados comentados se deduce claramente que los estudios comparativos de predicción y exploración/reducción representan líneas de investigación minoritarias respecto a los estudios comparativos de clasificación. Como veremos a continuación,

una de las líneas de investigación de esta tesis versa sobre la aplicación de RNA a la predicción del tiempo de supervivencia. Por otro lado, hemos realizado un estudio comparativo (Sesé, Palmer, Montaño, Jiménez, Sospedra y Cajal, 2001), en el ámbito de la Psicometría, entre PCA clásico y redes neuronales orientadas a la reducción de la dimensionalidad: regla de Oja (1982, 1989), mapas autoorganizados (Kohonen, 1982) y redes autoasociativas lineales y no lineales (Kramer, 1991). Los resultados ponen de manifiesto la utilidad de las RNA especialmente en aquellos casos en que se introducen relaciones complejas o no lineales entre las variables implicadas.

1.3. Líneas de investigación de la tesis.

A partir de los resultados obtenidos en el estudio bibliométrico realizado, iniciamos dos líneas de investigación cuyo objetivo consistía en aplicar modelos de redes neuronales a dos campos en los que nuestro equipo había estado trabajando en los últimos años: análisis de datos aplicado a conductas adictivas (Calafat, Amengual, Farrés y Palmer, 1985; Calafat, Amengual, Mejias, Borrás y Palmer, 1989; Palmer, Amengual y Calafat, 1992; Palmer, 1993a; Calafat, Amengual, Palmer y Mejias, 1994; Calafat, Amengual, Palmer y Saliba, 1997) y análisis de supervivencia (Palmer, 1985; Palmer, 1993b; Palmer y Cajal, 1996; Palmer y Losilla, 1998). Como se ha comentado, la aplicación de las RNA a estos dos campos es claramente deficitaria a juzgar por el número de trabajos realizados. Por otra parte, iniciamos una tercera línea de investigación sobre un tema propio del campo de las RNA que constituye el principal inconveniente en la utilización de este tipo de tecnología como herramienta en el análisis de datos: el estudio del efecto de las variables de entrada en una red MLP. En la figura 16 se muestran las líneas de investigación que forman la presente tesis y que, a continuación, pasamos a describir.

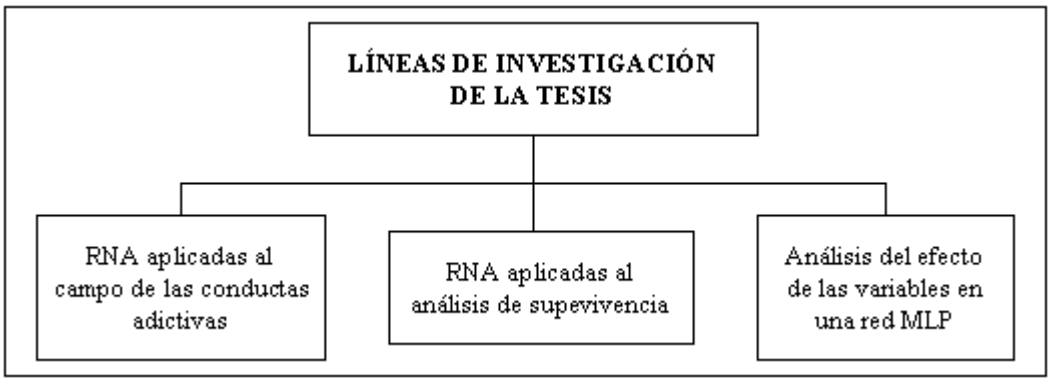


Figura 16. Líneas de investigación de la tesis.

1.3.1. RNA aplicadas al campo de las conductas adictivas.

Hemos podido comprobar a partir del estudio bibliométrico que las principales áreas de aplicación de las RNA en nuestra base de datos son la medicina, la ingeniería y la biología. También se ha observado que las aplicaciones en el campo de las ciencias del comportamiento aún son incipientes. Dentro de este campo, el uso y abuso de sustancias comprende un conjunto de conductas complejas que son iniciadas, mantenidas y modificadas por una variedad de factores conocidos y desconocidos. El tipo de función o relación que se establece entre la conducta adictiva y los factores que la explican no se puede reducir a una simple relación lineal de "causa-efecto" (Buscema, 1997, 1998). Creemos que las propiedades que caracterizan las RNA como, por ejemplo, la capacidad de aprender funciones complejas no conocidas *a priori*, el manejo de multitud de variables de entrada sin suponer un aumento significativo en la complejidad del modelo y la ausencia de restricciones sobre los datos, se adaptan perfectamente a las características de este tipo de fenómenos sociales.

En este sentido, el Centro de Investigación Semeion de las Ciencias de la Comunicación (Roma, Italia), fundado y dirigido por Massimo Buscema, ha sido pionero en la aplicación de las RNA para la prevención y predicción de la conducta adictiva (Buscema, 1999, 2000). Los investigadores de dicho centro han construido diferentes modelos de red con el fin de predecir el consumo de drogas –sobre todo heroína— (Buscema, 1995; Buscema, Intraligi y Bricolo, 1998; Maurelli y Di Giulio, 1998; Speri et al., 1998), extraer las características prototípicas del sujeto adicto (Buscema, Intraligi y Bricolo, 1998) y así, determinar el tratamiento más adecuado en función de esas características (Massini y Shabtay, 1998). Una revisión exhaustiva de las aplicaciones realizadas mediante RNA en este campo se puede encontrar en nuestro trabajo titulado *¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones* (Palmer y Montaño, 1999) (ver apartado 2.1., pág. 117). En este trabajo también se realiza una introducción al campo de las RNA explicando conceptos fundamentales tales como el funcionamiento de la neurona artificial, tipos de arquitecturas y algoritmos de aprendizaje, ventajas respecto a los modelos estadísticos clásicos y aplicaciones generales.

Siguiendo la línea de investigación iniciada por el equipo de Buscema, nos proponemos como objetivos llevar a cabo la aplicación práctica de una red neuronal para la

predicción del consumo de éxtasis (MDMA) en la población de jóvenes europeos e identificar los factores de riesgo asociados al consumo de esta sustancia mediante la aplicación de un análisis de sensibilidad. Más concretamente, se trata de construir un modelo de red neuronal que a partir de las respuestas de los sujetos a un cuestionario, sea capaz de discriminar entre quién consume éxtasis y quién no. Este trabajo que lleva por título *Predicción del consumo de éxtasis a partir de redes neuronales artificiales* (Palmer, Montaño y Calafat, 2000) (ver apartado 2.2., pág. 133), junto al anterior trabajo comentado, han sido revisados por Buscema el cual es miembro del comité editorial de la revista *Adicciones* donde han sido publicados ambos trabajos.

El consumo de éxtasis y otros derivados de las feniletilaminas ha experimentado un aumento significativo en los últimos años, provocando una gran alarma social (Plan Nacional sobre Drogas, 2000). Contrariamente a lo que se pensaba en un inicio, los recientes casos de muerte provocada por la ingesta de este tipo de sustancias, ha demostrado su alto grado de toxicidad, además de otros efectos negativos sobre el sistema nervioso. Con nuestro estudio, se intenta averiguar si las RNA pueden ser empleadas en un futuro como herramientas de apoyo al profesional dedicado a la prevención del consumo de este tipo de sustancias.

La recogida de datos se ha realizado en colaboración con la ONG IREFREA (Instituto y Red Europea para el Estudio de los Factores de Riesgo en la Infancia y la Adolescencia). Esta organización tiene por objetivo la creación de conocimiento empírico sobre la realidad social en relación al consumo de drogas en la población de jóvenes, prestando especial atención al estudio de los factores de riesgo que determinan este tipo de conductas. IREFREA está extendida en siete países pertenecientes a la Comunidad Europea (Alemania, Austria, Italia, España, Francia, Grecia y Portugal). También participan en sus proyectos instituciones y equipos investigadores de otros países como la *Public Health Sector* de la Universidad John Moore de Liverpool (Inglaterra), el *Instituto de Medicina Legale*, el CSST/CREDIT (*Centre Hopitalier Universitaire de Nice, Centre de Soins Specialisés pour Toxicomanes*) en Niza, el SPI (*Socialpädagogisches Institut*) en Berlín y el *Institute Sozial und Gesundherst* en Viena.

La muestra inicial estaba compuesta por 1900 jóvenes (con edad media de 22.42 años y desviación estándar de 4.25) pertenecientes a cinco países: España, Francia, Holanda, Italia y Portugal. A cada uno de los sujetos se le aplicó un cuestionario en el que se

estudiaban diversos aspectos bio-psico-sociales y conductas relacionadas con el consumo de drogas. Los principales resultados obtenidos con este conjunto de datos se encuentran en Calafat, Stocco et al. (1998), Calafat, Bohrn et al. (1999) y Calafat, Juan et al. (2000).

Para los intereses de nuestra investigación, seleccionamos de la muestra inicial 148 consumidores de éxtasis y 148 no consumidores de éxtasis. El grupo de consumidores se caracterizaba por ser consumidores habituales de éxtasis –consumían éxtasis más de una vez al mes. En general, los sujetos que formaban esta categoría eran además consumidores de otras sustancias como marihuana, cocaína, anfetaminas y heroína. Por su parte, el grupo de no consumidores que había servido como grupo control se caracterizaba por no haber consumido nunca éxtasis ni ninguna otra sustancia ilegal.

Con el objeto de determinar las características predictoras del consumo de éxtasis, fueron seleccionados 25 items del cuestionario original. Los items se podían agrupar en cinco categorías temáticas:

a) Demografía, relaciones con los padres y creencias religiosas
b) Ocio
c) Consumo
d) Opinión sobre el éxtasis
e) Personalidad

Las áreas exploradas por este conjunto de items coinciden en gran medida con los principios de la *Squashing Theory*, enfoque desarrollado por Buscema (1995) y encaminado a la predicción de la conducta adictiva, mediante un modelo de red neuronal, a partir del registro de diversas medidas biológicas, psicológicas y sociológicas.

La aplicación de la red neuronal en la discriminación de consumidores y no consumidores de éxtasis a partir de los valores obtenidos en las variables predictoras, se ha realizado siguiendo cinco fases:

1) Selección de las variables relevantes y preprocesamiento de los datos
2) Creación de los conjuntos de aprendizaje, validación y test
3) Entrenamiento de la red neuronal
4) Evaluación del rendimiento de la red neuronal
5) Análisis de sensibilidad.

Cabe destacar que la red neuronal utilizada fue una arquitectura MLP compuesta por 25 neuronas de entrada, dos neuronas ocultas y una de salida. El algoritmo de aprendizaje fue el *backpropagation* con una tasa de aprendizaje igual a 0.3 y un factor momento igual a 0.8. La evaluación del modelo se ha realizado a partir de los índices de sensibilidad, especificidad y eficacia, y del análisis de curvas ROC (*Receiver Operating Characteristics*) (Swets, 1973, 1988). Como veremos más adelante, el análisis de curvas ROC constituye una forma eficaz de evaluar el rendimiento de un modelo orientado a la discriminación de dos categorías (en este caso, consumo o no consumo de éxtasis), cuando el valor que proporciona es cuantitativo.

Por último, intentando superar una de las críticas más importantes que se han lanzado contra el uso de las RNA –esto es, la dificultad para comprender la naturaleza de las representaciones internas generadas por la red--, hemos aplicado un procedimiento denominado análisis de sensibilidad dirigido a determinar el efecto o importancia de cada variable predictora sobre el consumo de éxtasis. Para ello, se fija el valor de todas las variables de entrada a un determinado valor y se va variando el valor de una de ellas a lo largo de todo su rango, con el objeto de observar el efecto que tiene sobre la salida de la red. Con ello, pretendemos identificar los factores de riesgo asociados al consumo de éxtasis. Como veremos posteriormente, este procedimiento ha ido evolucionando hacia un método denominado por nosotros análisis de sensibilidad numérico que permite superar algunas de las limitaciones observadas en otros métodos propuestos con anterioridad.

1.3.2. RNA aplicadas al análisis de supervivencia.

El análisis de supervivencia está formado por un conjunto de técnicas, cuya utilidad se encuentra en estudios longitudinales en los cuales se desea analizar el tiempo transcurrido hasta la ocurrencia de un determinado suceso, previamente definido.

El término “supervivencia” se debe al hecho de que inicialmente estas técnicas se utilizaron para estudiar eventos terminales como la muerte de los individuos, si bien se ha generalizado su uso y actualmente se aplican para el estudio del tiempo transcurrido desde un determinado momento hasta que se produce cualquier tipo de evento, como el tiempo hasta la curación de una enfermedad desde el momento de inicio de una terapia,

o el tiempo que un sujeto tarda en presentar una recaída desde el momento en que desaparece el trastorno.

Para llevar a cabo un análisis de supervivencia se necesitan, como mínimo, los valores de dos variables para cada sujeto: una variable que defina el tiempo transcurrido y otra variable que defina el estado final del sujeto, es decir, si el sujeto ha realizado o no el cambio de estado de interés antes del momento de cierre del estudio.

Para definir el tiempo transcurrido se requiere el punto temporal de origen, que representa el momento en el que el sujeto entra a formar parte del estudio (que puede ser distinto para cada sujeto), así como el punto temporal en el que el sujeto realiza el cambio de estado. En el supuesto de que al finalizar el estudio el sujeto todavía mantenga su estado inicial, el segundo momento temporal viene definido por la fecha de cierre del estudio, la cual debe ser la misma para todas las observaciones. Por su parte, se define el desenlace del estudio como la realización del cambio que constituye el paso del estado inicial en el que se encuentra el sujeto hasta el estado final.

La presencia de información incompleta o censurada constituye una característica fundamental en los datos de supervivencia que hace difícil su manejo mediante los métodos estadísticos (Allison, 1995) y modelos de RNA (Ohno-Machado, 1997) convencionales. Así, por ejemplo, el modelo de regresión lineal múltiple, comúnmente utilizado para la estimación de variables de naturaleza continua, no sería adecuado para la estimación del tiempo de supervivencia debido a que no es capaz de manejar la información parcial proporcionada por los datos incompletos. De forma similar, una red MLP convencional no es capaz de predecir el tiempo de supervivencia manejando información incompleta. Posiblemente este sea el motivo por el cual apenas existen trabajos que traten el análisis de supervivencia con modelos RNA.

Un dato incompleto es aquella observación que, formando parte de la muestra de estudio, no ha realizado el cambio de estado. Ello se puede deber, fundamentalmente, a dos motivos: que en un momento determinado del seguimiento el sujeto se pierda (es retirado del estudio porque presenta algún tipo de incompatibilidad) o desaparezca del estudio (no acude al seguimiento), o bien que al finalizar el estudio el sujeto todavía no haya realizado el cambio. En cualquiera de los dos casos, la información que se tiene es el tiempo transcurrido hasta la última observación, aunque este tiempo no defina el tiempo transcurrido hasta el cambio porque éste no se haya producido. Por esta razón se

dice que un dato incompleto contiene tan sólo información parcial. La información que proporciona un dato incompleto es que durante un determinado tiempo no se produjo el cambio, y por tanto desconocemos cuándo se produciría (si se llegara a producir) en el futuro.

La figura 17 muestra el esquema de los tiempos de participación y los tipos de datos y de cambios que se estudian en el análisis de supervivencia.

En la figura se puede observar que los datos censurados son incompletos en el lado derecho del período de seguimiento, siendo debido a que el estudio finaliza o el sujeto se pierde en un momento dado. Generalmente este tipo de datos se denominan datos censurados por la derecha (Kleinbaum, 1996). Aunque pueden existir datos censurados por la izquierda o datos censurados por intervalo, únicamente consideraremos los censurados por la derecha ya que la mayoría de datos de supervivencia son de este tipo.

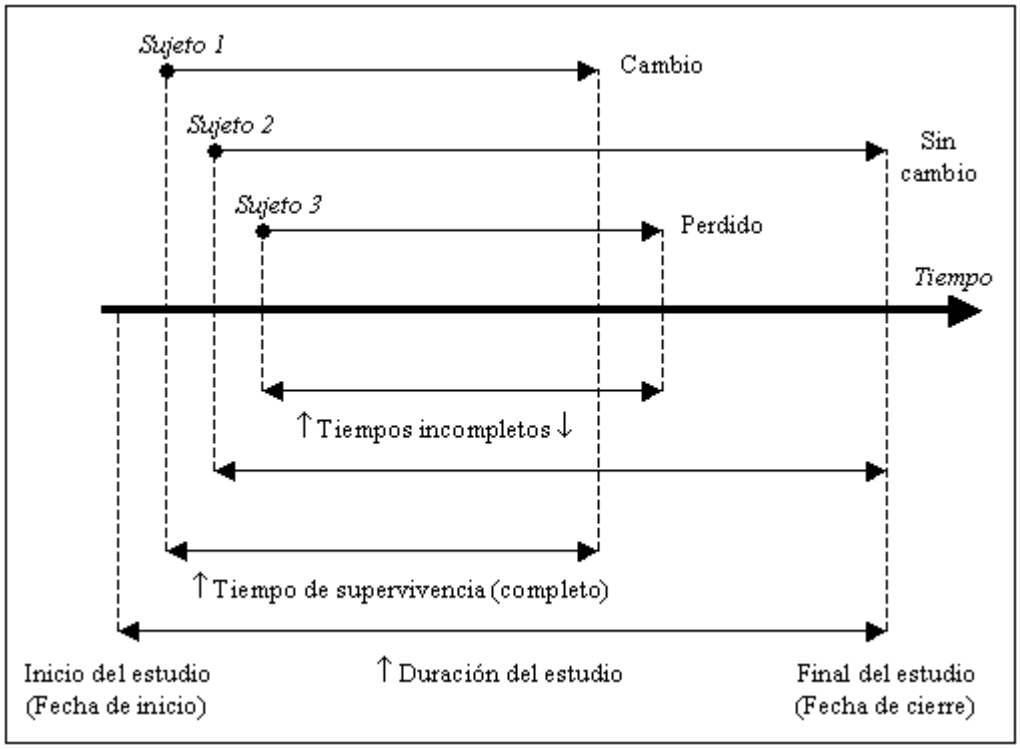


Figura 17. Esquema de los tipos de datos en el análisis de supervivencia (Palmer y Losilla, 1998).

En los datos de supervivencia también se pueden utilizar variables dependientes del tiempo, esto es, variables cuyos valores pueden cambiar a lo largo del período de observación.

En el contexto del análisis de supervivencia, se establece la variable aleatoria no negativa T, como el tiempo transcurrido hasta la ocurrencia del desenlace. Esta variable queda perfectamente definida por medio de su función de probabilidad, en el caso de tiempo discreto, y por su función de densidad de probabilidad en el caso de tiempo continuo. A partir de esta función se definen las funciones de supervivencia y de riesgo. De esta forma, la función de densidad del tiempo transcurrido hasta la realización de un cambio, proporciona, en cada instante o intervalo diferencial de tiempo, la probabilidad de que un sujeto realice el cambio de interés en el estudio. Por su parte, la función de supervivencia S(t) es la probabilidad acumulada de que el tiempo transcurrido, hasta que un sujeto realice el cambio, sea superior a un determinado momento temporal t. Es decir, el valor de esta función en un tiempo t expresa la probabilidad de que un sujeto no haya realizado el cambio hasta ese instante. Por último, la función de riesgo h(t) es, de forma aproximada, la probabilidad de realizar un cambio en un instante o intervalo diferencial de tiempo, condicionado a que no se haya producido el cambio previamente. De hecho, la función de riesgo es una tasa de cambio porque su denominador está expresado en unidades persona-tiempo, indicando la intensidad (número) de cambios por unidad de tiempo.

En este punto, hemos contemplado tres posibles estrategias para realizar el análisis de datos de supervivencia (ver figura 18).

Como primera estrategia tenemos el análisis mediante modelos estadísticos. Esta estrategia ha sido la más utilizada en el contexto del análisis de datos de supervivencia.

Desde una perspectiva puramente descriptiva, los métodos estadísticos no paramétricos actuarial (*life table*) y Kaplan-Meier permiten estimar las tres funciones relevantes mencionadas anteriormente (densidad, supervivencia y riesgo).

La elección de un método u otro dependerá de cómo estén recogidos los datos. Si se dispone de la información individual, el método de Kaplan-Meier será el elegido, mientras que si se dispone de la información agrupada en intervalos, el método a utilizar será el método actuarial. El método de Kaplan-Meier realiza una estimación para cada instante t en el que se ha producido algún desenlace, mientras que el método actuarial realiza una estimación para cada intervalo temporal que tengamos definido.

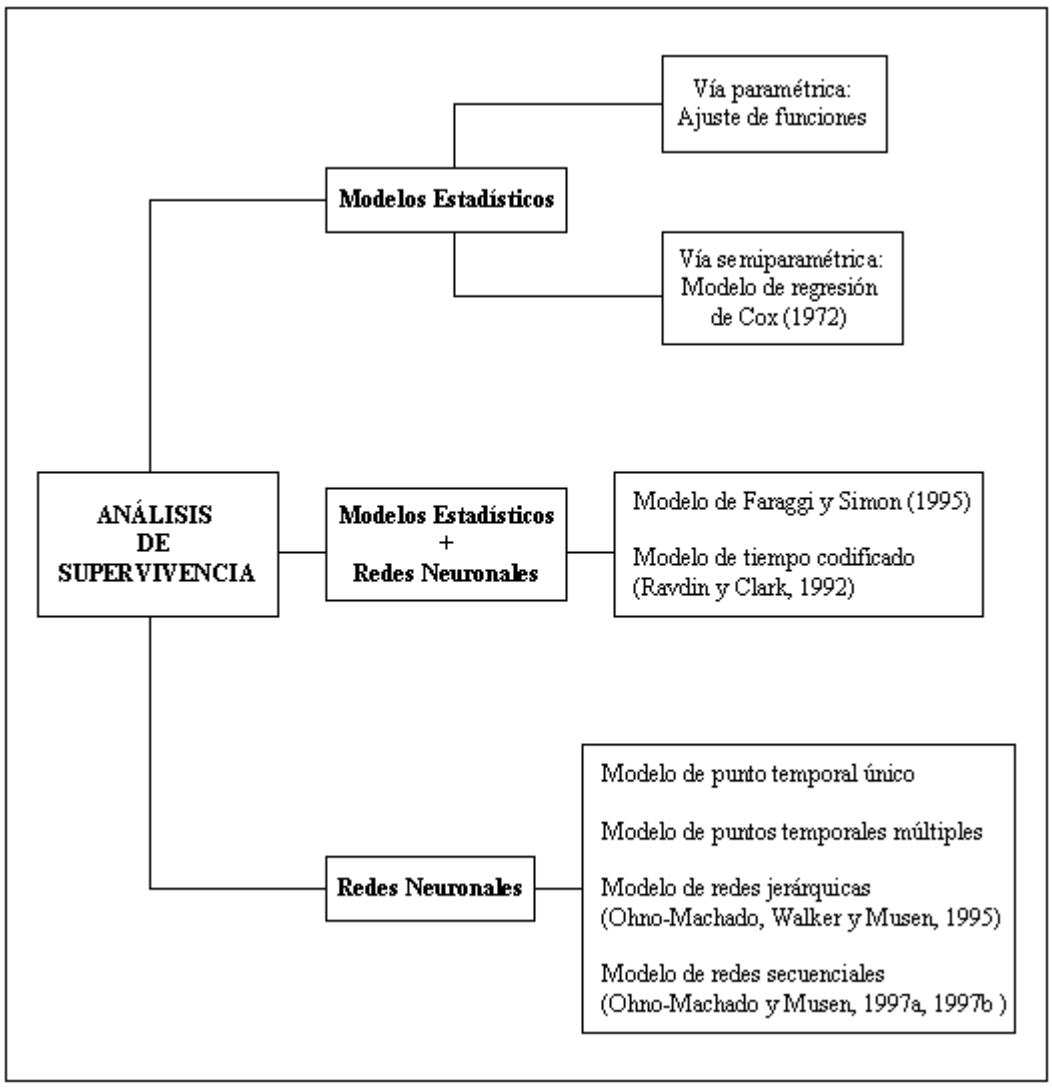


Figura 18. Estrategias para realizar el análisis de datos de supervivencia.

La otra diferencia básica entre ambos métodos de estimación consiste en la definición que hacen de los sujetos expuestos al riesgo: en el método de Kaplan-Meier se consideran expuestos al riesgo todos los sujetos que entran en el instante t, mientras que en el método actuarial los datos incompletos que ocurren en el intervalo sólo cuentan la mitad. El método de Kaplan-Meier es, sin duda, el que permite describir de forma más completa y precisa el proceso de cambio, si bien el método actuarial permite resumir el proceso de forma más compacta.

Por último, la forma habitual de describir gráficamente la función de supervivencia es por medio del denominado “gráfico de escalera”, en el cual la altura de cada peldaño representa la disminución de la función al pasar de un instante o intervalo al siguiente, y

la anchura de los peldaños proporcionan la distancia desde un desenlace al siguiente (Kaplan-Meier) o la amplitud de intervalo (actuarial) (ver figura 19).

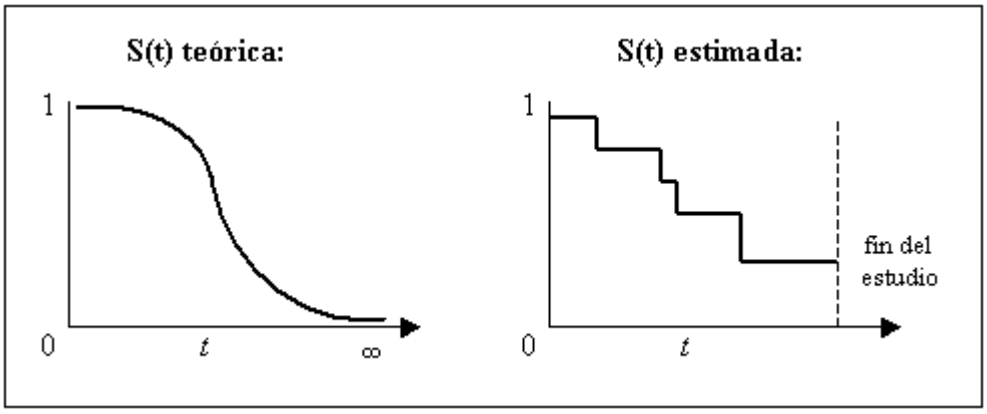


Figura 19. Función de supervivencia teórica y estimada.

Desde una perspectiva explicativa, se puede optar por la vía paramétrica o por la vía semiparamétrica. El análisis paramétrico requiere conocer la función de densidad de probabilidad de la variable tiempo (por ejemplo, distribución exponencial, distribución log-normal o distribución Weibull), motivo por el cual es poco utilizado en las ciencias del comportamiento, en las cuales, como en nuestras investigaciones, se utiliza el análisis semiparamétrico, que no requiere dicha información.

El modelo de regresión de riesgos proporcionales (*proportional hazards model*), conocido habitualmente como modelo de regresión de Cox (Cox, 1972), es el modelo más utilizado en el contexto del análisis no paramétrico y relaciona la función de riesgo con las hipotéticas variables explicativas por medio de la expresión:

$$h(t, X) = h_0(t)\exp(B'X) \qquad (14)$$

donde, h(t, X) es la tasa de riesgo de realizar el cambio en el instante t de un sujeto con un determinado patrón de valores X en las variables explicativas en dicho instante; $h_0(t)$ es la función de riesgo de línea base que depende sólo del tiempo y que representa el riesgo de cambio en cada instante t de un sujeto ficticio con valor 0 en todas las variables explicativas, y B' es el vector de coeficientes asociados al vector X de variables explicativas.

Este modelo puede describirse (Allison, 1984) como semiparamétrico o parcialmente paramétrico. Es paramétrico ya que especifica un modelo de regresión con una forma funcional específica; es no paramétrico en cuanto que no especifica la forma exacta de la distribución de los tiempos de supervivencia.

La interpretación práctica del efecto de una variable explicativa se realiza a partir del exponente $\exp(B)$, que es el factor por el cual se multiplica el "riesgo relativo" cuando dicha variable se incrementa en una unidad (manteniendo constantes las demás variables).

Debido a la existencia de datos incompletos, los parámetros del modelo de Cox no pueden ser estimados por el método ordinario de máxima verosimilitud al ser desconocida la forma específica de la función arbitraria de riesgo. Cox (1975) propuso un método de estimación denominado verosimilitud parcial (*partial likelihood*), siendo las verosimilitudes condicionales y marginales casos particulares del anterior. El método de verosimilitud parcial se diferencia del método de verosimilitud ordinario en el sentido de que mientras el método ordinario se basa en el producto de las verosimilitudes para todos los individuos de la muestra, el método parcial se basa en el producto de las verosimilitudes de todos los cambios ocurridos.

Para estimar los coeficientes B en el modelo de Cox, en ausencia de conocimiento de $h_0(t)$, éste propuso la siguiente función de verosimilitud:

$$L(B) = \prod_{i=1}^{K} \frac{\exp(X'B)}{\sum_{R_i} \exp(X'B)} \qquad (15)$$

donde K hace referencia a los momentos en los que se produce algún cambio y R_i hace referencia a todos los sujetos expuestos al riesgo en el instante t_i.

Esta expresión L(B) no es una verdadera función de verosimilitud ya que no puede derivarse como la probabilidad de algún resultado observado bajo el modelo de estudio, si bien, como indica Cox (1975), puede tratarse como una función de verosimilitud ordinaria a efectos de realizar estimaciones de B. Dichas estimaciones son consistentes (Cox, 1975; Tsiatis, 1981) y eficientes (Efron, 1977).

El logaritmo de la función de verosimilitud parcial viene dado por:

$$\mathrm{LnL(B)} = \sum_{i=1}^{k} (S_i B) - \sum_{i=1}^{k} \left[d_i \mathrm{Ln} \left[\sum_{R_i} (\exp(B'x)) \right] \right] \qquad (16)$$

siendo S_i la suma de los valores de la variable concomitante para todos los d_i sujetos que realizan el cambio en el instante t_i. El tercer sumatorio se realiza para todos los sujetos expuestos al riesgo, designados como R_i, en el instante t_i.

Las estimaciones máximo verosímiles de B son estimaciones que maximizan la función LnL(B). El proceso de maximización se realiza tomando la derivada parcial de la función LnL(B) con respecto a cada parámetro B_i que compone el modelo y resolviendo el siguiente sistema de ecuaciones:

$$\frac{\mathrm{LnL(B)}}{B_i} = 0 \qquad (17)$$

Para ello, se sigue un proceso iterativo mediante la aplicación del método de Newton-Raphson (Palmer, 1993b).

El modelo de Cox contiene implícitamente dos supuestos. El primer supuesto asume una relación multiplicativa entre la función arbitraria de riesgo y la función log-lineal de las variables explicativas. Este es el supuesto de proporcionalidad que debe su nombre al hecho de que dos sujetos cualesquiera i y j presentarán, en todo momento, riesgos proporcionales (Allison, 1984). De esta forma, el ratio de los riesgos de ambos sujetos debe permanecer constante:

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t)\exp(B'X_i)}{h_0(t)\exp(B'X_j)} = \frac{\exp(B'X_i)}{\exp(B'X_j)} = \text{constante} \qquad (18)$$

donde la constante puede depender de las variables explicativas pero no del tiempo. Dicho con otras palabras, el efecto de las diferentes variables explicativas debe mantenerse constante a lo largo del período de seguimiento y, por tanto, ese efecto es independiente del tiempo. El supuesto de proporcionalidad se puede verificar mediante

diversos procedimientos (Blossfeld, Hamerle y Mayer, 1989; Kleinbaum, 1996): representaciones gráficas (gráfico log-log, gráfico de observados vs. esperados), pruebas de bondad de ajuste, introducción de variables dependientes del tiempo y análisis de residuales.

En el caso que una variable explicativa incumpla el supuesto de proporcionalidad, habitualmente se excluye del modelo la variable y ésta se trata como variable de estrato (Marubini y Valsecchi, 1995; Parmar y Machin, 1995). De esta forma, se obtienen las funciones de riesgo y supervivencia de forma separada para cada grupo formado a partir de la variable estratificada, mientras que los coeficientes de regresión son iguales para todos los grupos o estratos. Blossfeld y Rohwer (1995) proponen una estrategia alternativa a la estratificación, que consiste en incluir en el modelo una variable dependiente del tiempo que refleje el efecto de interacción entre la variable explicativa y la tasa instantánea de riesgo, con lo cual se corrige automáticamente el incumplimiento del supuesto.

El segundo supuesto asume un efecto log-lineal de las variables explicativas sobre la función de riesgo. Es decir, las variables explicativas actúan sobre la función de riesgo de forma multiplicativa.

Un aspecto importante del modelo de Cox radica en que éste se puede utilizar para realizar predicciones sobre el proceso de cambio. Más concretamente, para los propósitos de nuestra investigación, nos proponemos utilizar el modelo de Cox para predecir la función de supervivencia, con unos determinados valores en las variables explicativas. La función de supervivencia para un sujeto dado, se puede obtener mediante el modelo de Cox a través de la siguiente expresión:

$$S(t, X) = S_0(t)^{\exp(B'X)} \tag{19}$$

donde S(t, X) es la función de supervivencia en el instante t de un sujeto con un determinado patrón de valores X en las variables explicativas en dicho instante y $S_0(t)$ es la función de supervivencia de línea base. $S_0(t)$ viene dada por:

$$S_0(t) = \exp(H_0(t)) \tag{20}$$

siendo $H_0(t)$ la función acumulada de riesgo de línea base.

Como segunda estrategia tenemos el uso de una red neuronal en conjunción con un modelo estadístico para ser aplicados a datos de supervivencia. Los modelos propuestos por Faraggi y Simon (1995) y Ravdin y Clark (1992) son ejemplos de este tipo de estrategia.

El modelo propuesto por Faraggi y Simon (1995) se basa en utilizar el método de estimación de máxima verosimilitud parcial para estimar los parámetros de una red MLP. Consideremos un Perceptrón compuesto de una capa de entrada con N neuronas, una capa oculta con L neuronas y una capa de salida con una sola neurona k. Teniendo en cuenta que la función de activación de las neuronas ocultas es la sigmoidal logística y la función de activación de la neurona de salida en la lineal, para un patrón p de entrada $X_p : x_{p1},...,x_{pi},...x_{pN}$, la salida de la neurona de salida k se puede representar mediante la siguiente expresión:

$$y_{pk} = \theta_k + \sum_{j=1}^{L} v_{jk} f(w_{ij} x_{pi} + \theta_j) = \theta_k + \sum_{j=1}^{L} \frac{v_{jk}}{1 + \exp(-w_{ij} x_{pi} + \theta_j)} \qquad (21)$$

Recordemos que el modelo de Cox relaciona la función de riesgo con las hipotéticas variables explicativas por medio de la expresión (14):

$$h(t, X) = h_0(t)\exp(B'X) \qquad (14)$$

El vector de parámetros B son estimados maximizando la función de verosimilitud parcial mediante la expresión (15):

$$L(B) = \prod_{i=1}^{K} \frac{\exp(X'B)}{\sum_{R_i} \exp(X'B)} \qquad (15)$$

usando el método de Newton-Raphson (Palmer, 1993b).

Consideremos sustituir la función lineal $B'X$ por la salida de la red y_{pk} en la expresión de la función de riesgo de Cox. El modelo de riesgos proporcionales se convierte en:

$$h(t, X) = h_0(t)\exp(y_{pk}) \tag{22}$$

y la función a maximizar que permite estimar los parámetros de la red W se convierte en:

$$L(W) = \prod_{i=1}^{K} \frac{\exp\left(\sum_{j=1}^{L} \frac{v_{jk}}{1+\exp(-w_{ij}x_{pi}+\theta_j)}\right)}{\sum_{R_i}\exp\left(\sum_{j=1}^{L} \frac{v_{jk}}{1+\exp(-w_{ij}x_{pi}+\theta_j)}\right)} \tag{23}$$

A la hora de aplicar este procedimiento debemos tener en cuenta, por un lado, que el peso umbral de la neurona de salida θ_k de la red no queda incluido en el modelo de Cox. Por otro lado, la utilización de la función de activación lineal sobre la neurona de salida, permite que la salida de la red esté situada en el eje real al igual que $B'X$, y no en el intervalo (0, 1) que sería el caso de haber utilizado la función sigmoidal logística.

Bajo esta propuesta, las estimaciones máximo verosímiles de los parámetros de la red neuronal no se obtienen mediante la aplicación del algoritmo de aprendizaje *backpropagation* sino que se obtienen usando el método de Newton-Raphson para maximizar la función de verosimilitud parcial definida. De esta forma, el uso de funciones de máxima verosimilitud para estimar los parámetros de la red, permite la utilización de pruebas y procedimientos estadísticos clásicos para evaluar la red neuronal.

Esta forma de construir modelos de red neuronal se puede extender a otros modelos que incorporan datos censurados, tales como el modelo de tiempo de fracaso acelerado (Prentice y Kalbfleisch, 1979) y el modelo de Buckley-James (Buckely y James, 1979).

El modelo propuesto por Ravdin y Clark (1992), denominado modelo de tiempo codificado, se basa en utilizar una red MLP codificando el tiempo de seguimiento como una variable de entrada o explicativa. La salida de la red realiza predicciones acerca de la probabilidad de cambio en un momento temporal dado (ver figura 20).

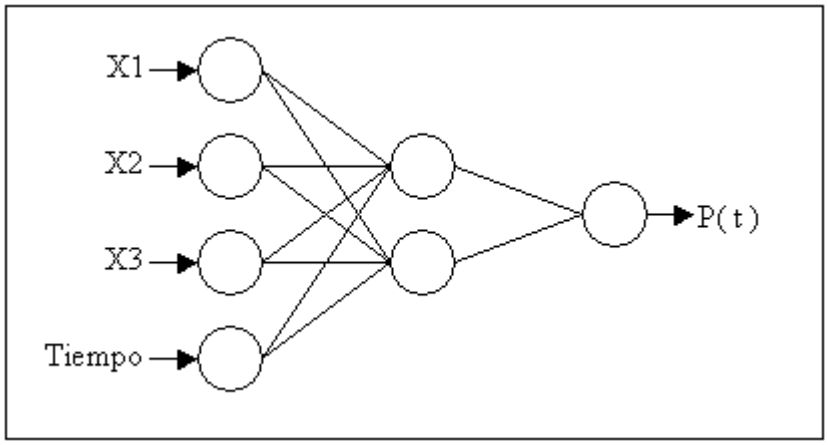


Figura 20. Modelo de tiempo codificado.

Para aplicar este modelo, en primer lugar, se establecen intervalos de tiempo con igual tasa de cambio utilizando el estimador Kaplan-Meier. Por ejemplo, se podrían crear intervalos de tiempo en los que hubiese un decremento constante en la función de supervivencia del 10%. A continuación, se transforma el vector de datos originales perteneciente a cada sujeto en un conjunto de vectores siguiendo el esquema de la figura 21.

Se puede observar que el vector original está compuesto por los valores en las variables explicativas (por ejemplo, X1, X2 y X3), el intervalo de tiempo del cambio o de último seguimiento y el estatus del sujeto asociado a ese momento. Para cada vector generado, se presenta como entrada a la red el valor de las variables explicativas junto al intervalo de tiempo t; la salida representa el estatus del sujeto en ese momento t que puede ser 0 ó 1 (no realizar el cambio o realizar el cambio de estado, respectivamente). Si el sujeto ha realizado el cambio, en el intervalo de tiempo en el que se ha producido el cambio y en los intervalos de tiempo sucesivos se proporciona como estatus o salida de la red un 1. Si el sujeto ha sido censurado antes de un intervalo de tiempo dado, entonces no se presenta ningún vector para éste y los sucesivos intervalos de tiempo (Ravdin y Clark, 1992). Bajo este esquema también es posible introducir variables dependientes del tiempo pudiendo proporcionar diferentes valores para un mismo sujeto en función del intervalo de tiempo en el que se encuentre.

Con el objeto de corregir el sesgo provocado por la presencia de datos censurados sobre la tasa de cambio en los últimos intervalos de tiempo, Ravdin y Clark (1992) proponen recurrir de nuevo al método Kaplan-Meier. Así, se realiza para cada intervalo una selección aleatoria de datos, equilibrando el número de vectores asociados a cambio y

no cambio de estado. Esta medida implica perder información debido a que tenemos que ignorar registros para alcanzar el citado equilibrio.

Dato completo:

vector original	(X1, X2, X3, variables explicativas	5, intervalo de tiempo del cambio	1) estatus
vectores resultantes	(X1, X2, X3,	1,	0)
	(X1, X2, X3,	2,	0)
	(X1, X2, X3,	3,	0)
	(X1, X2, X3,	4,	0)
	(X1, X2, X3,	5,	1)
	(X1, X2, X3,	6,	1)
	(X1, X2, X3,	7,	1)
	.	.	.
	.	.	.

Dato incompleto:

vector original	(X1, X2, X3, variables explicativas	4, intervalo de tiempo de último seguimiento	0) estatus
vectores resultantes	(X1, X2, X3,	1,	0)
	(X1, X2, X3,	2,	0)
	(X1, X2, X3,	3,	0)
	(X1, X2, X3,	4,	0)

Figura 21. Esquema de transformación de los datos originales.

Este modelo tiene la ventaja de ser muy simple, ya que con una sola arquitectura se pueden realizar predicciones para todos los intervalos de tiempo considerados. Por otra parte, hemos podido observar que es capaz de manejar datos censurados e introducir variables dependientes del tiempo. Sin embargo, implica una cantidad considerable de procesamiento de datos al tener que duplicar selectivamente los casos que tienen tiempos de supervivencia altos.

De Laurentiis y Ravdin han realizado un exhaustivo estudio de simulación en el que se pone de manifiesto una serie de ventajas en el modelo de tiempo codificado respecto al

modelo de Cox (De Laurentiis y Ravdin, 1994a; De Laurentiis y Ravdin, 1994b). En primer lugar, las RNA no están sujetas al cumplimiento del supuesto de proporcionalidad. En segundo lugar, las RNA detectan automáticamente funciones lineales y cuadráticas y términos de interacción de primer y segundo orden sin necesidad de ser explicitados en el modelo. Por último, el área bajo la curva ROC (*Receiver Operating Characteristic*) (Swets, 1973, 1988) muestra una mayor capacidad de discriminación entre cambio o no cambio por parte de las RNA.

Finalmente, tenemos la perspectiva del análisis de datos de supervivencia mediante la utilización de un modelo exclusivamente neuronal. Se puede trazar una evolución desde los modelos más simples e intuitivos hasta los modelos más avanzados y sofisticados que permiten superar las limitaciones de los primeros (Palmer y Losilla, 1999).

La primera propuesta, denominada modelo de punto temporal único, consiste en proporcionar como salida una estimación del estatus del sujeto (0 ó 1) en un momento específico del seguimiento a partir de los valores de un conjunto de variables predictoras. Para ello, se utiliza una red MLP como la mostrada en la figura 22.

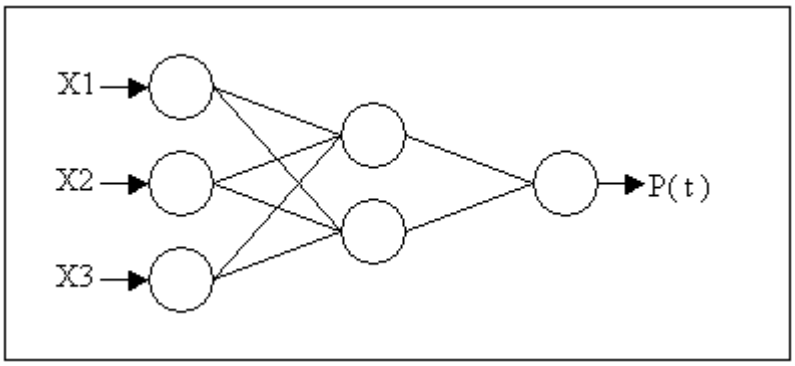


Figura 22. Modelo de punto temporal único.

Esta forma de análisis es análoga al modelo de regresión logística y se reduce a un análisis transversal de los datos en el que no se tiene en cuenta la dimensión temporal. El modelo de punto temporal único ha sido muy utilizado en el ámbito de la medicina donde en general se pretende predecir si se ha dado el suceso de interés –fallecimiento, recidiva, intervención quirúrgica, etc.--, o no hasta el momento en que finaliza el estudio (Ebell, 1993; Clark, Hilsenbeck, Ravdin, De Laurentiis y Osborne, 1994; Lapuerta, Azen, Labree, 1995; Frye, Izenberg, Williams y Luterman, 1996; Lippmann y Shahian, 1997; Burke, Hoang, Iglehart y Marks, 1998; Lundin, Lundin, Burke,

Toikkanen, Pylkkänen y Joensuu, 1999; Kehoe, Lowe, Powell y Vincente, 2000; Faraggi, LeBlanc y Crowley, 2001). La limitación de este modelo radica en que no es capaz de manejar datos censurados ni variables dependientes del tiempo. De Laurentiis y Ravdin (1994a) proponen dos posibles estrategias para manejar datos censurados. Se puede optar por ignorar los casos que son censurados, solución adoptada por la mayoría de aplicaciones revisadas, o se puede estimar el resultado de los datos censurados mediante el modelo de Cox.

Un modelo que permite tener en cuenta el proceso de cambio es el modelo de puntos temporales múltiples. Hace uso de una red MLP cuyas neuronas de entrada reciben el vector de variables predictoras y cuyas neuronas de salida proporcionan una estimación del estatus del sujeto (0 ó 1) en diferentes intervalos de tiempo (ver figura 23).

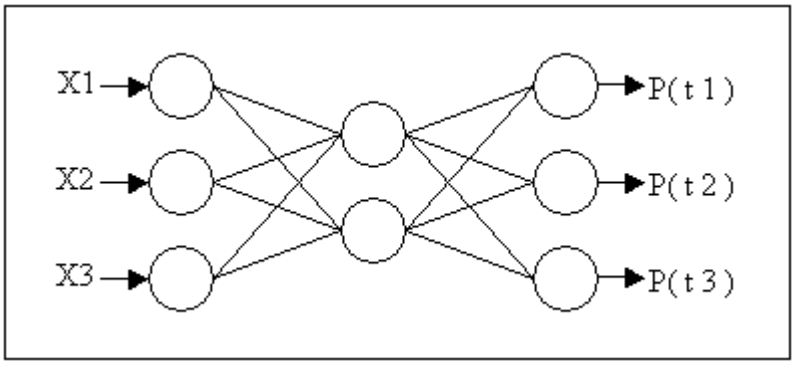


Figura 23. Modelo de puntos temporales múltiples.

De esta forma, cada neurona de salida k es entrenada para proporcionar el valor 1 cuando el sujeto no ha realizado el cambio hasta el intervalo de tiempo k, y el valor 0 cuando el sujeto ha realizado el cambio en el momento k o en un momento anterior. Este modelo tampoco puede manejar datos censurados ni variables dependientes del tiempo. Aquellos sujetos cuyo estatus no es conocido en todos los intervalos temporales considerados, no pueden ser utilizados por la red neuronal ya que es necesario aportar un valor concreto (0 ó 1) de salida para cada momento. Ohno-Machado ha realizado varios estudios comparativos entre el modelo de puntos temporales múltiples y el modelo de Cox a partir de conjuntos de datos que no contienen datos censurados por perdida o desaparición del sujeto (Ohno-Machado y Musen, 1995; Ohno-Machado, Walker y Musen, 1995; Ohno-Machado, 1997). La comparación se realizó en función

del valor obtenido con los índices de eficacia, sensibilidad, especificidad, valor predictivo positivo y negativo, mostrando ambos modelos un rendimiento similar.

De Laurentiis y Ravdin (1994a) han realizado dos propuestas, de forma similar al caso anterior, para solucionar el problema de los casos censurados en el modelo de puntos temporales múltiples. En primer lugar, se puede optar por estimar el resultado de los datos censurados mediante el modelo de regresión de Cox. En segundo lugar, durante la fase de entrenamiento, se puede asignar a las neuronas de salida cuyo intervalo de tiempo está censurado, un valor específico que sea reconocido por la red y que permita desactivar esa neurona, de forma que no participe en el entrenamiento de la red. Esta opción se encuentra implementada en el programa simulador de RNA *Neural Works Professional II* (NeuralWare, 1995) y fue aplicada sobre un conjunto de datos simulados obteniendo un rendimiento superior respecto al modelo de Cox (De Laurentiis y Ravdin, 1994a).

Finalmente, Ohno-Machado ha propuesto dos modelos de red diseñados para el análisis de supervivencia con datos censurados y variables dependientes del tiempo: el modelo de redes jerárquicas (Ohno-Machado, Walker y Musen, 1995) y el modelo de redes secuenciales (Ohno-Machado y Musen, 1997a; Ohno-Machado y Musen, 1997b).

El modelo de redes jerárquicas (Ohno-Machado, Walker y Musen, 1995) consiste en una arquitectura jerárquica de redes neuronales del tipo MLP que predicen la supervivencia mediante un método paso a paso (ver figura 24).

De este modo, cada red neuronal se encarga de dar como salida la probabilidad de supervivencia en un intervalo de tiempo determinado, proporcionando el modelo general la supervivencia para el primer intervalo, después para el segundo intervalo y así sucesivamente. Si el sujeto ha realizado el cambio en el intervalo de tiempo t se proporciona como valor de salida deseado el valor 0 para la red correspondiente al intervalo de tiempo t y los sucesivos intervalos. Si el sujeto no ha realizado el cambio hasta el momento t se proporciona como valor de salida deseado el valor 1. Este modelo permite, por una parte, el manejo de datos censurados debido a que cada red neuronal se construye a partir de aquellos sujetos para los cuales se tiene información hasta el intervalo de tiempo correspondiente. Por ejemplo, los datos de un sujeto al que se le haya realizado el seguimiento hasta el tercer intervalo considerado, serán usados en las redes correspondientes al primer, segundo y tercer intervalo, pero no en las redes

correspondientes a los siguientes intervalos de tiempo. Por otra parte, se pueden utilizar variables dependientes del tiempo debido a que cada red neuronal puede recibir un valor de entrada diferente respecto a las variables explicativas para un mismo sujeto. Por último, se pueden crear curvas de supervivencia tanto a nivel individual como a nivel de grupo, a partir de las probabilidades de supervivencia proporcionadas por el modelo general para los sucesivos intervalos de tiempo.

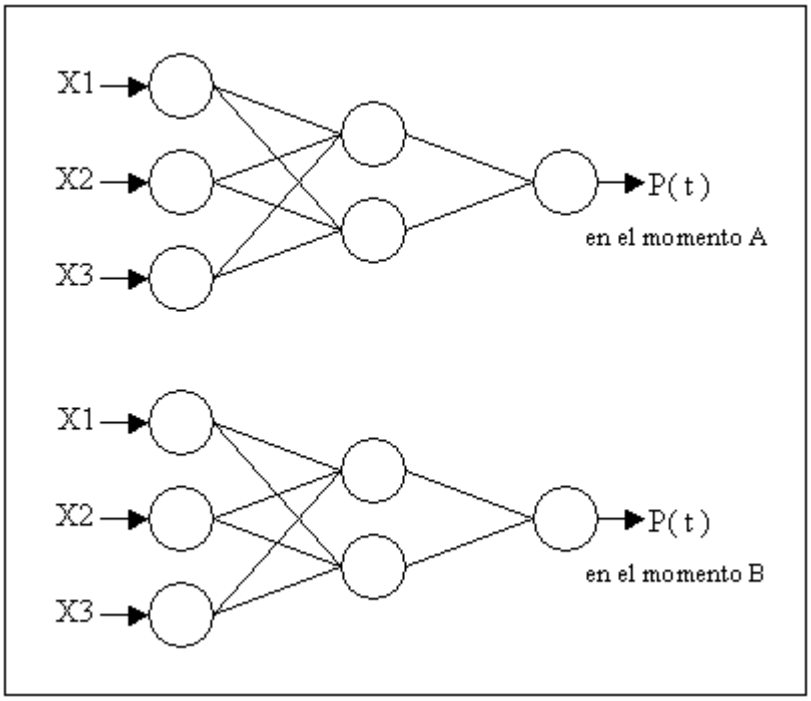


Figura 24. Modelo de redes jerárquicas.

El modelo de redes secuenciales (Ohno-Machado y Musen, 1997a; Ohno-Machado y Musen, 1997b) supone una ampliación respecto al modelo de redes jerárquicas. Tomando como base el esquema de trabajo del modelo de redes jerárquicas, en el modelo de redes secuenciales la predicción realizada por una red neuronal para un intervalo de tiempo, actúa a su vez como variable explicativa o de entrada en otra red dedicada a la predicción de otro intervalo anterior o posterior. De esta forma, el intervalo correspondiente a la primera red neuronal actúa como intervalo informativo y el intervalo correspondiente a la segunda red neuronal actúa como intervalo informado (ver figura 25).

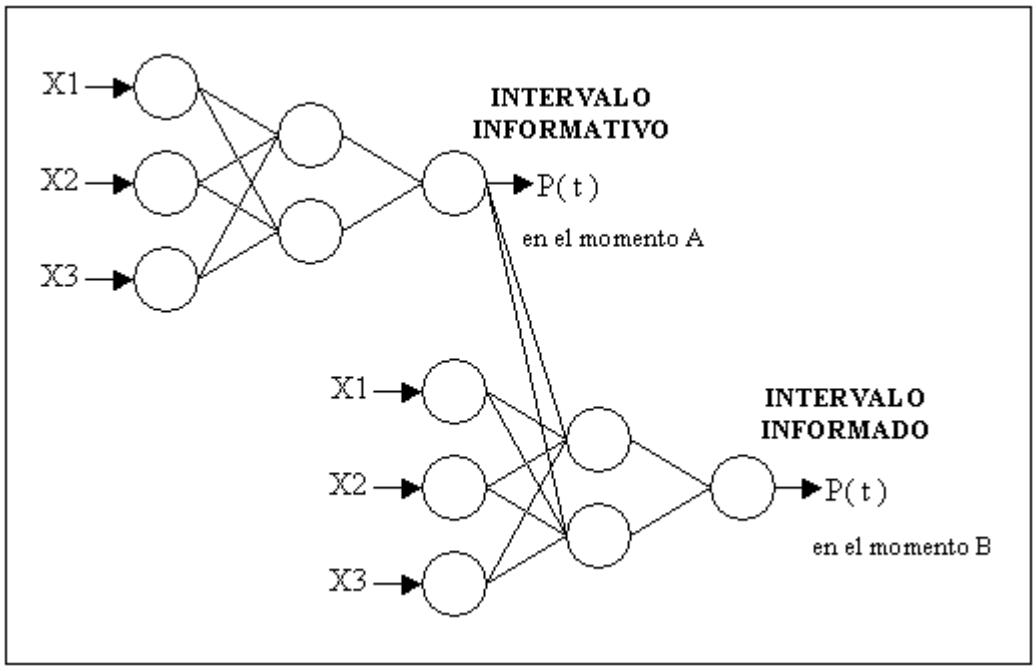


Figura 25. Modelo de redes secuenciales.

Con esta estrategia, se pretende modelar explícitamente la dependencia temporal que existe entre las predicciones realizadas en los diferentes intervalos de tiempo. Como consecuencia, las predicciones realizadas se ajustarán mejor a la realidad y las curvas de supervivencia serán asintóticamente decrecientes a diferencia de los modelos anteriormente propuestos.

Ohno-Machado (1996) realizó un estudio comparativo entre modelos de redes jerárquicas, redes secuenciales y modelo de Cox a partir de dos conjuntos de datos pertenecientes al ámbito de la medicina (supervivencia en sujetos con enfermedad coronaria y supervivencia en sujetos con SIDA). Como criterio de comparación se utilizaron los valores del área bajo la curva ROC (Swets, 1973, 1988) y la prueba de Hosmer-Lemeshow (Hosmer y Lemeshow, 1980). Como resultado, tanto las redes jerárquicas como las redes secuenciales mostraron un rendimiento superior frente al modelo de Cox en la mayoría de intervalos de tiempo considerados. Por su parte, las redes secuenciales no mostraron un mejor rendimiento en función de la prueba de Hosmer-Lemeshow respecto al modelo de redes jerárquicas, aunque en la mayoría de intervalos de tiempo si obtuvieron un mejor rendimiento en función de las curvas ROC. Por último, se demostró la utilidad de las redes jerárquicas y secuenciales para obtener curvas de supervivencia asintóticamente decrecientes.

Siguiendo la línea de investigación iniciada por Ohno-Machado, en nuestro trabajo titulado *Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo* (Palmer y Montaño, 2002) (ver apartado 2.4., pág. 165), se realiza una comparación en cuanto a capacidad de predicción entre modelos de RNA jerárquicos y secuenciales, y el modelo de regresión de Cox.

Más concretamente, nos planteamos como objetivo comprobar las siguientes hipótesis: a) el modelo de redes jerárquicas presenta un rendimiento superior en cuanto a predicción frente al modelo de regresión de Cox, b) el modelo de redes secuenciales supone una mejora en rendimiento respecto al modelo de redes jerárquicas, y c) los modelos de red proporcionan curvas de supervivencia más ajustadas a la realidad que el modelo de Cox.

Para ello, se han utilizado los datos procedentes de una serie de estudios realizados por el equipo de McCusker (McCusker et al., 1995; McCusker, Bigelow, Frost et al., 1997; McCusker, Bigelow, Vickers-Lahti et al., 1997) en la Universidad de Massachusetts (la matriz de datos, denominada uis.dat, se puede obtener en la sección *Survival Analysis* de la siguiente dirección URL: http://www-unix.oit.umass.edu/~statdata). El objetivo de estos estudios fue comparar diferentes programas de intervención diseñados para la reducción del abuso de drogas en una muestra de 628 toxicómanos. En nuestro estudio, se han utilizado nueve variables predictoras. La variable de respuesta es el tiempo en días transcurrido desde el inicio del estudio hasta la recaída del sujeto en el consumo de drogas. Tras definir ocho intervalos de tiempo en los que la probabilidad de supervivencia disminuye de forma aproximadamente constante a medida que avanza el seguimiento, se procedió a generar los modelos de red y el modelo de Cox.

Las redes neuronales jerárquicas y secuenciales se han construido siguiendo el esquema descrito anteriormente utilizando el algoritmo de aprendizaje de gradientes conjugados (Battiti, 1992). Así, el modelo de redes jerárquicas se compone de ocho redes MLP, cada una centrada en dar como salida la probabilidad de supervivencia en uno de los intervalos de tiempo creados, mientras que el modelo de redes secuenciales se compone de 56 redes MLP como resultado de cruzar entre sí las ocho redes del modelo jerárquico. Por su parte, el modelo de Cox se ha generado, bajo la perspectiva de

comparación de modelos, mediante un método de selección paso a paso hacia atrás basado en la razón de verosimilitud.

La eficacia en cuanto a predicción se ha comparado sobre un grupo de test a partir de medidas de resolución y calibración. Ambas medidas son independientes y complementarias, debido a que hay modelos que tienen una buena resolución y una mala calibración, mientras que hay modelos a los que les sucede lo contrario (Ohno-Machado, 1996).

La resolución la definimos como la capacidad de discriminar por parte del modelo entre sujetos que realizan el cambio y sujetos que no realizan el cambio de estado. La resolución se ha medido a partir del análisis de curvas ROC (*Receiver Operating Characteristics*) (Swets, 1973, 1988), generalmente utilizado para evaluar el rendimiento de pruebas diagnósticas cuyo objetivo es discriminar entre dos tipos de sujetos (normalmente, “sanos” y “enfermos”).

Aplicado a nuestro estudio, la curva ROC consiste en la representación gráfica del porcentaje de verdaderos positivos (sensibilidad = S) en el eje de ordenadas, contra el porcentaje de falsos positivos (1-especificidad = NE) en el eje de abcisas, para diferentes puntos de corte aplicados sobre el valor cuantitativo estimado por el modelo. Este valor oscila entre 0 y 1, donde 0 significa cambio y 1 significa no cambio de estado. Esta codificación permite interpretar los valores estimados por los modelos de red como la función de supervivencia del sujeto en cada intervalo de tiempo considerado. Los verdaderos positivos son sujetos que presentan el cambio en un momento dado y han sido clasificados correctamente, mientras que los falsos positivos son sujetos que no presentan el cambio en un momento dado y han sido clasificados incorrectamente como sujetos con cambio de estado.

La capacidad de discriminar entre dos categorías por parte de una red neuronal ha sido evaluada tradicionalmente mediante la obtención de una tabla de contingencia 2x2 y calculando, a partir de ella, el porcentaje de clasificaciones correctas o alguna medida de acuerdo o asociación (índices Kappa, Phi, etc.) y, en el contexto clínico, calculando los índices de sensibilidad, especificidad y eficacia. Esto es debido, en parte, a que la mayoría de programas simuladores de RNA se limitan a proporcionar este tipo de información una vez entrenada la red neuronal. Ejemplos de investigaciones que utilizan este tipo de estrategia son las de Somoza y Somoza (1993), Speight, Elliott,

Jullien, Downer y Zakzrewska (1995) y Penny y Frost (1997). Sin embargo, estos procedimientos dependen de la aplicación de un determinado punto de corte sobre el valor cuantitativo (en general, entre 0 y 1) proporcionado por la red. Así, por ejemplo, el uso de un punto de corte bajo dará lugar a una alta sensibilidad a costa de una baja especificidad, y con un punto de corte alto sucederá lo contrario (Turner, 1978). El uso de curvas ROC permite superar esta limitación, debido a que resumen en un solo gráfico la información derivada de todas las posibles tablas de contingencia 2x2 resultantes de la aplicación de distintos puntos de corte (Swets, 1986).

La curva ROC refleja el grado de solapamiento de las estimaciones del modelo en los dos grupos de interés (cambio/no cambio) (ver figura 26). Cuando el solapamiento es total (modelo inútil), la curva ROC recorre la diagonal positiva del gráfico, ya que para cualquier punto de corte S = NE. Cuando el solapamiento es nulo (test perfecto), la curva ROC recorre los bordes izquierdo y superior del gráfico, ya que para cualquier punto de corte, o bien S = 1, o bien NE = 0, existiendo algún punto de corte en el que S = 1 y NE = 0. En la práctica el solapamiento de valores para un grupo u otro será parcial, generando curvas ROC intermedias entre las dos situaciones planteadas (Weinstein y Fineberg, 1980).

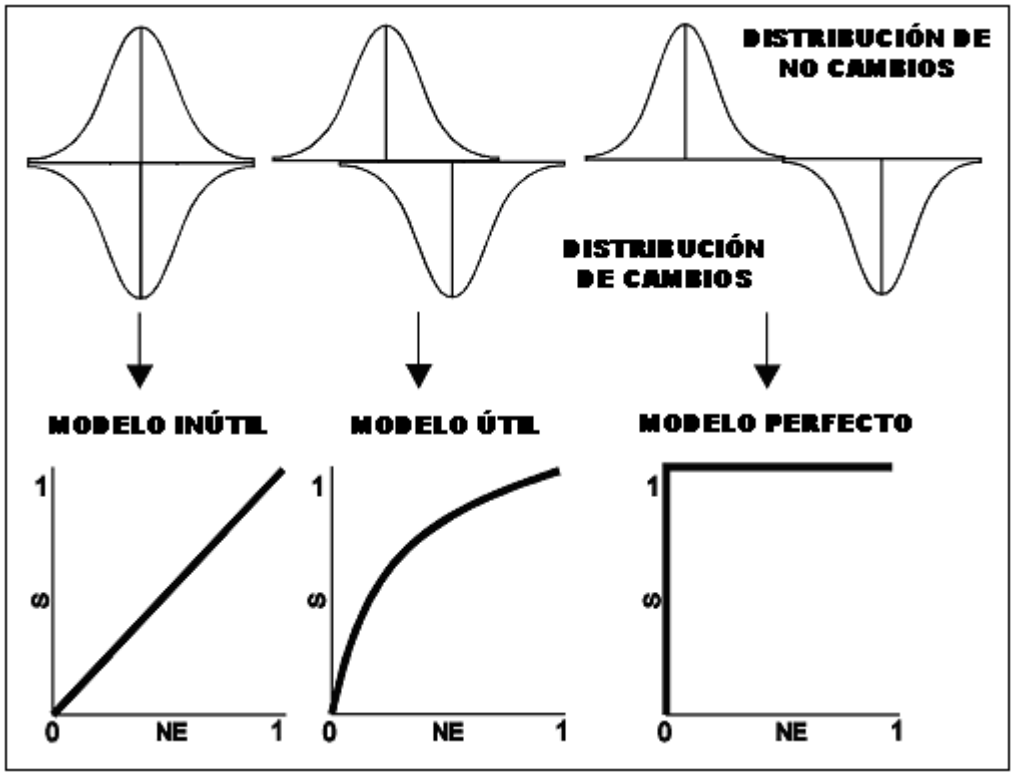


Figura 26. Correspondencia entre solapamiento de las distribuciones y la curva ROC.

En nuestra investigación, la obtención de la curva ROC se ha realizado mediante una aproximación no paramétrica consistente en la representación de la curva empírica. Esta aproximación es la más adecuada cuando se trabaja con datos obtenidos en una escala de medida cuantitativa (Burgueño, García-Bastos y Gónzalez-Buitrago, 1995).

En este tipo de análisis, la medida de resumen más utilizada es el área total bajo la curva ROC (AUC, *Area Under ROC-Curve*). Esta medida se interpreta como la probabilidad de clasificar correctamente un par de sujetos –uno que ha realizado el cambio y otro que no--, seleccionados al azar, fluctuando su valor entre 0.5 y 1. El AUC de un modelo inútil es 0.5, reflejando que al ser utilizado clasificamos correctamente un 50% de individuos, idéntico porcentaje al obtenido utilizando simplemente el azar. Por el contrario, el AUC de un modelo perfecto es 1, ya que permite clasificar sin error el 100% de sujetos.

En nuestro estudio, el cálculo del AUC y su error estándar se ha realizado mediante las expresiones no paramétricas descritas por Hanley y McNeil (1982), las cuales han sido aplicadas a los modelos de red –redes jerárquicas y redes secuenciales— y al modelo de Cox para cada uno de los intervalos de tiempo creados. La comparación entre los modelos a partir del AUC se ha realizado mediante la prueba z de Hanley y McNeil (1983) que viene dada por:

$$z = \frac{AUC_{modelo1} - AUC_{modelo2}}{\sqrt{SE^2_{modelo1} + SE^2_{modelo2} - 2 \cdot r \cdot SE_{modelo1} \cdot SE_{modelo2}}} \qquad (24)$$

donde $AUC_{modelo1}$ y $AUC_{modelo2}$ son las AUC de los dos modelos comparados, $SE_{modelo1}$ y $SE_{modelo2}$ son los errores estándar de las AUC de los dos modelos comparados y r representa la correlación entre las dos AUC comparadas.

Por su parte, la calibración la definimos como el ajuste de las predicciones realizadas por el modelo respecto al resultado real. La calibración se ha medido a partir de la prueba χ^2 de Hosmer-Lemeshow (Hosmer y Lemeshow, 1980), comúnmente utilizada como medida de la bondad de ajuste en los modelos de regresión logística (Glantz, 1990). Para ello, ordenamos ascendentemente las estimaciones realizadas por el modelo y se divide la muestra en diez grupos en función del valor de los deciles. A

continuación, se obtiene la suma de valores observados n_i y la suma de valores esperados por el modelo e_i para cada grupo g creado. Para determinar la significación estadística de la diferencia entre valores observados y valores esperados se utiliza la siguiente prueba:

$$\chi^2 = \sum_{g=1}^{G} \frac{(n_i - e_i)^2}{e_i} \tag{25}$$

con g – 2 grados de libertad.

1.3.3. Análisis del efecto de las variables en una red Perceptrón multicapa.

El estudio del efecto o importancia de las variables de entrada en una red MLP es uno de los aspectos más críticos en la utilización de las RNA orientadas al análisis de datos, debido a que el valor de los parámetros obtenidos por la red no tienen una interpretación práctica a diferencia de un modelo de regresión clásico. Como consecuencia, las RNA se han presentado al usuario como una especie de "caja negra" a partir de las cuales no es posible analizar el papel que desempeña cada variable de entrada en la predicción realizada.

La limitación presentada por las RNA en cuanto a su aspecto explicativo nos ha llevado a realizar un estudio exhaustivo de los trabajos que han tratado el problema. Se ha podido observar que desde finales de los años 80 se han propuesto diferentes métodos dirigidos a interpretar lo aprendido por una red MLP. Siguiendo el esquema presentado en la figura 27, estos métodos interpretativos se pueden dividir en dos tipos de metodologías: análisis basado en la magnitud de los pesos y análisis de sensibilidad.

La descripción de este conjunto de métodos se puede encontrar en nuestros trabajos *Redes neuronales artificiales: abriendo la caja negra* (Montaño, Palmer y Fernández, 2002) (ver apartado 2.5, pág. 175) y *Numeric sensitivity analysis applied to feedforward neural networks* (Montaño y Palmer, en revisión) (ver apartado 2.6., pág. 195).

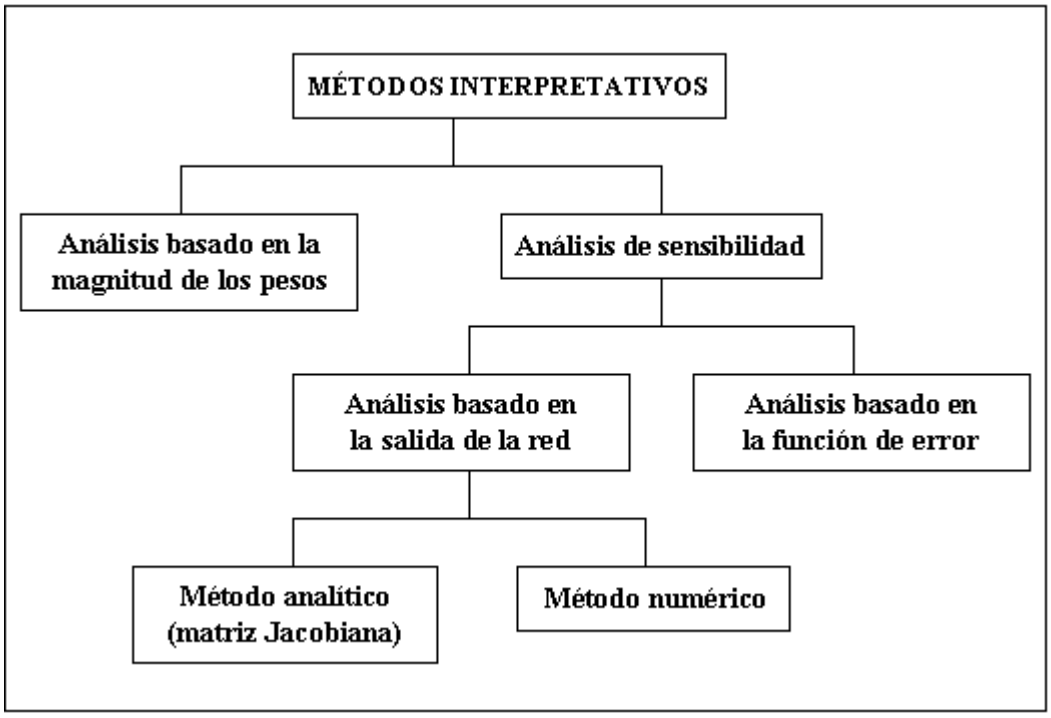


Figura 27. Esquema de los métodos interpretativos propuestos.

De forma resumida, el análisis basado en la magnitud de los pesos agrupa aquellos procedimientos que se basan exclusivamente en los valores almacenados en la matriz estática de pesos con el propósito de determinar la influencia relativa de cada variable de entrada sobre cada una de las salidas de la red. Se han propuesto diferentes ecuaciones basadas en la magnitud de los pesos, una de las más representativas es la presentada por Garson (1991). Por su parte, Tchaban, Taylor y Griffin (1998) han presentado una variante interesante de este tipo de análisis denominada *weight product* que incorpora a la información proporcionada por los pesos, el cociente entre el valor de la variable de entrada x_i y el valor de la salida y_k.

El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida y_k o en el error cometido debido al cambio que se produce en una entrada x_i. El análisis de sensibilidad aplicado sobre una función de error se puede realizar de diversas formas. Una modalidad consiste en ir variando el valor de una de las variables de entrada a lo largo de todo su rango mediante la aplicación de pequeños incrementos o perturbaciones, mientras se mantienen los valores originales de las demás variables de entrada (Frost y Karri, 1999). También se puede optar por restringir la entrada de interés a un valor fijo (por ejemplo, el valor promedio) para todos los patrones (Smith, 1993) o

eliminar directamente esa entrada (Wang, Jones y Partridge, 2000). En cualquiera de estos casos, el procedimiento se basa en concluir que la entrada es importante en la predicción realizada por el modelo, si el error aumenta sensiblemente ante el cambio provocado.

El método más habitual de realizar el análisis de sensibilidad consiste en estudiar el efecto de las entradas directamente en la salida estimada por la red. Esto se puede realizar de forma similar al análisis aplicado sobre una función de error, fijando el valor de todas las variables de entrada a su valor medio, excepto una variable sobre la cual se añade ruido o pequeños incrementos. A continuación, se miden los cambios producidos en la salida de la red. Este procedimiento lo hemos aplicado en la predicción del consumo de éxtasis, como hemos visto anteriormente (Palmer, Montaño y Calafat, 2000). Un procedimiento que goza de una mejor fundamentación matemática se basa en la obtención de la matriz Jacobiana mediante el cálculo de las derivadas parciales de las salidas y_k con respecto a las entradas x_i, esto es, $\frac{\partial y_k}{\partial x_i}$; el cual constituye la versión analítica del análisis de sensibilidad. Por este motivo, hemos denominado a este procedimiento análisis de sensibilidad analítico o método ASA (*Analytic Sensitivity Analysis*) (Montaño, Palmer y Fernández, 2002).

Los métodos descritos han demostrado su utilidad en determinadas tareas de predicción, sin embargo, cuentan con una serie de limitaciones. El análisis basado en la magnitud de los pesos no ha demostrado ser sensible a la hora de ordenar las variables de entrada en función de su importancia sobre la salida en los trabajos de simulación realizados (Sarle, 2000). El análisis de sensibilidad sobre el error o sobre la salida consistente en añadir incrementos o perturbaciones se basa en la utilización de variables de entrada cuya naturaleza es cuantitativa, ya que no sería del todo correcto añadir incrementos a variables nominales, esto es, variables que toman valores discretos (Hunter, Kennedy, Henry y Ferguson, 2000). Por último, el método ASA basado en el cálculo de la matriz Jacobiana, parte del supuesto de que todas las variables implicadas en el modelo son cuantitativas (Sarle, 2000). Este supuesto limita el número de campos de aplicación de las RNA en las Ciencias del Comportamiento donde es muy común el manejo de variables discretas (por ejemplo, género: 0 = varón, 1 = mujer).

En los dos trabajos citados (Montaño, Palmer y Fernández, 2002; Montaño y Palmer, en revisión) presentamos un nuevo método, denominado análisis de sensibilidad numérico o método NSA (*Numeric Sensitivity Analysis*), el cual permite superar las limitaciones comentadas de los métodos anteriores. Este nuevo método se basa en el cálculo de las pendientes que se forman entre entradas y salidas, sin realizar ningún supuesto acerca de la naturaleza de las variables y respetando la estructura original de los datos.

Una de las limitaciones a las que nos enfrentamos a la hora de estudiar el rendimiento de los diferentes métodos interpretativos, es que no existen programas informáticos que implementen tales métodos. Como consecuencia, nos propusimos como primer objetivo de esta línea de investigación, la creación de un programa que permitiera superar esta limitación. Para ello, diseñamos el programa *Sensitivity Neural Network 1.0*, el cual simula el comportamiento de una red MLP asociada a la regla de aprendizaje *backpropagation* y, como novedad, incorpora los siguientes métodos interpretativos para el análisis del efecto de las variables de entrada: método de Garson, análisis de sensibilidad analítico y análisis de sensibilidad numérico. Complementariamente a estos métodos numéricos, este programa también incorpora un procedimiento de visualización que muestra la representación gráfica de la función subyacente que la red neuronal ha aprendido entre cada par de variables de entrada y salida. *Sensitivity Neural Network* está siendo revisado por la revista *Behavior Research: Methods, Instruments, & Computers* con el trabajo titulado *Sensitivity Neural Network: an artificial neural network simulator with sensitivity analysis* (Palmer, Montaño y Fernández, en revisión) (ver apartado 2.7., pág. 213). Una descripción detallada del funcionamiento del programa se encuentra en el anexo 2 (pág. 299): *Sensitivity Neural Network 1.0: User's Guide*.

Una vez creado el programa *Sensitivity Neural Network*, nos propusimos como segundo objetivo realizar un estudio comparativo acerca del rendimiento de los siguientes métodos interpretativos: análisis basado en la magnitud de los pesos a través del método de Garson (1991), método *weight product* (Tchaban, Taylor y Griffin, 1998), análisis de sensibilidad basado en el cálculo del incremento observado en la función RMC error (Raíz cuadrada de la Media Cuadrática del error), método ASA y método NSA.

Para realizar el estudio comparativo se generaron mediante simulación ocho matrices de datos compuesta cada una de ellas por 1000 registros y cuatro variables con rango entre

0 y 1. Las tres primeras variables de cada matriz (X1, X2 y X3) actúan como variables predictoras o variables de entrada a la red, mientras que la última variable (Y) es una función de las variables predictoras y actúa como variable de salida. El valor del coeficiente de correlación de Pearson entre las variables predictoras oscila entre 0 y 0.40. En todos los casos, la variable X1 no tiene ningún tipo de contribución o efecto en la salida Y de la red, seguida de la variable X2 con un efecto intermedio y la variable X3 que presenta el mayor efecto sobre la salida de la red. A fin de analizar el comportamiento de los diferentes métodos dependiendo del tipo de variable implicada, se manipuló en cada una de las matrices la naturaleza de las variables de entrada, dando lugar a cuatro tipos de matriz: variables cuantitativas, variables discretas binarias, variables discretas politómicas y variables cuantitativas y discretas (binarias y politómicas). Cuando las variables son cuantitativas, la relación entre entradas y salida se estableció a través de funciones no lineales (sigmoidales y exponenciales), cuando las variables son binarias y politómicas la relación se ha establecido a través de los índices de asociación Phi y V de Cramer, respectivamente.

La creación de las redes neuronales y la posterior aplicación de los métodos interpretativos se realizó mediante el programa *Sensitivity Neural Network*. La descripción detallada de esta investigación se encuentra en los trabajos anteriormente comentados: *Redes neuronales artificiales: abriendo la caja negra* (Montaño, Palmer y Fernández, 2002) (ver apartado 2.5, pág. 175) y *Numeric sensitivity analysis applied to feedforward neural networks* (Montaño y Palmer, en revisión) (ver apartado 2.6., pág. 195).

1.4. Objetivos e hipótesis.

A modo de resumen, se exponen los objetivos e hipótesis que nos planteamos en la tesis, según la línea de investigación llevada a cabo.

1) En la aplicación de RNA al campo de las conductas adictivas, nos proponemos dos objetivos:
   a) La creación de una red neuronal capaz de discriminar entre sujetos consumidores y no consumidores de éxtasis a partir de las respuestas dadas a un cuestionario en la población de jóvenes europeos.

b) La identificación de los factores de riesgo asociados al consumo de éxtasis mediante la aplicación de un análisis de sensibilidad.

2) En la aplicación de RNA al análisis de supervivencia, nos proponemos realizar una comparación en cuanto a capacidad de predicción entre dos modelos de RNA (redes jerárquicas y redes secuenciales) y el modelo de regresión de Cox. Más concretamente, nos planteamos las siguientes hipótesis:
   a) El modelo de redes jerárquicas presenta un rendimiento superior en cuanto a predicción frente al modelo de regresión de Cox.
   b) El modelo de redes secuenciales supone una mejora en rendimiento respecto al modelo de redes jerárquicas.
   c) Los modelos de red proporcionan curvas de supervivencia más ajustadas a la realidad que el modelo de Cox.

3) En el estudio del efecto de las variables de entrada en una red MLP, nos proponemos dos objetivos:
   a) El diseño del programa *Sensitivity Neural Network 1.0*, el cual permita simular el comportamiento de una red MLP asociada a la regla de aprendizaje *backpropagation* e incorpore los siguientes métodos interpretativos para el análisis del efecto de las variables de entrada: método de Garson, el método ASA (*Analytic Sensitivity Analysis*) y el método NSA (*Numeric Sensitivity Analysis*).
   b) La realización de un estudio comparativo sobre el rendimiento de los siguientes métodos interpretativos: análisis basado en la magnitud de los pesos a través del método de Garson (1991), método *weight product* (Tchaban, Taylor y Griffin, 1998), análisis de sensibilidad basado en el cálculo del incremento observado en la función RMC error (Raíz cuadrada de la Media Cuadrática del error), método ASA y método NSA. Más concretamente, queremos comprobar si el método NSA desarrollado por nosotros, supera en rendimiento a los demás métodos analizados.

Referencias Bibliográficas

Ackley, D.H., Hinton, G.E. y Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science, 9*, 147-169.

Albus, J.S. (1975). New approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 220-227.

Alcain, M.D. (1991). Aspectos métricos de la información científica. *Ciencias de la Información (La Habana), diciembre*, 32-36.

Allison, P.D. (1984). *Event history analysis. Regression for longitudinal event data*. Beverly Hills, CA: Sage Pub.

Allison, P.D. (1995). *Survival analysis using the SAS system: a practical guide*. Cary, NC: SAS Institute Inc.

Amat, N. (1994). *La documentación y sus tecnologías*. Madrid: Pirámide.

Anderson, J.A., Silverstein, J.W., Ritz, S.A. y Jones, R.S. (1977). Distinctive features, categorical perception and probability learning: some applications of a neural model. *Psychological Review, 84*, 413-451.

Bahbah, A.G. y Girgis, A.A. (1999). Input feature selection for real-time transient stability assessment for artificial neural network (ANN) using ANN sensitivity analysis. En IEEE (Ed.), *Proceedings of the 21st International Conference on Power Industry Computer Applications* (pp. 295-300). Piscataway, NJ: IEEE.

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S. y Lewis, P.A. (1994). A study of the classification capabilities of neural networks using unsupervised learning: a comparison with k-means clustering. *Psychometrika, 59*(4), 509-525.

Baldi, P. y Hornik, K. (1989). Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks, 2*(1), 53-58.

Battiti, R. (1992). First and second order methods for learning: between steepest descent and Newton's method. *Neural Computation, 4*(2), 141-166.

Baxt, W.G. (1991). Use of an artificial neural network for the diagnosis of myocardial infarction. *Annals of Internal Medicine, 115*(11), 843-848.

Bertsekas, D.P. (1995). *Nonlinear programming*. Belmont, MA: Athena Scientific.

Bertsekas, D.P. y Tsitsiklis, J.N. (1996). *Neuro-dynamic programming*. Belmont, MA: Athena Scientific.

Biganzoli, E., Boracchi, P., Mariani, L. y Marubini, E. (1998). Feed-forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in Medicine, 17*(10), 1169-1186.

Bishop, C.M. (1994). Neural networks and their applications. *Review of Scientific Instruments, 65*(6), 1803-1832.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Blossfeld, H.P., Hamerle, A. y Mayer, K.U. (1989). *Event history analysis*. Hillsdale, NJ: LEA.

Blossfeld, H.P. y Rohwer, G. (1995). *Techniques of event history modeling*. Mahwah, NJ: Lawrence Erlbaum Associates, Pub.

Bourland, H. y Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics, 59*, 291-294.

Broomhead, D.S. y Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems, 2,* 321-355.

Buckely, J. y James, I. (1979). Linear regression with censored data. *Biometrika, 66*, 429-436.

Burgueño, M.J., García-Bastos, J.L. y Gónzalez-Buitrago, J.M. (1995). Las curvas ROC en la evaluación de las pruebas diagnósticas. *Medicina clínica, 104*, 661-670.

Burke, H.B., Hoang, A., Iglehart, J.D. y Marks, J.R. (1998). Predicting response to adjuvant and radiation therapy in patients with early stage breast carcinoma. *Cancer, 82*(5), 874-877.

Buscema, M. (1995). Squashing Theory: A prediction approach for drug behavior. *Drugs and Society, 8*(3-4), 103-110.

Buscema, M. (1997). A general presentation of artificial neural networks. I. *Substance Use & Misuse, 32*(1), 97-112.

Buscema, M. (1998). Artificial neural networks and complex systems. I. Theory. *Substance Use & Misuse, 33*(1), 1-220.

Buscema, M. (1999). Redes neuronales artificiales y toxicodependencias. *Adicciones, 11*(4), 295-297.

Buscema, M. (2000). Redes neuronales y problemas sociales. *Adicciones, 12*(1), 99-126.

Buscema, M., Intraligi, M. y Bricolo, R. (1998). Artificial neural networks for drug vulnerability recognition and dynamic scenarios simulation. *Substance Use & Misuse, 33*(3), 587-623.

Cajal, B., Jiménez, R., Losilla, J.M., Montaño, J.J., Navarro, J.B., Palmer, A., Pitarque, A., Portell, M., Rodrigo, M.F., Ruíz, J.C. y Vives, J. (2001). Las redes neuronales

artificiales en psicología: un estudio bibliométrico. *Metodología de las Ciencias del Comportamiento, 3*(1), 53-64.

Calafat, A., Amengual, M., Farrés, C. y Palmer, A. (1985). Life-style and drug use habits among secondary school students. *Bulletin on Narcotics*, 37(2 y 3), 113-123.

Calafat, A., Amengual, M., Mejias, G., Borrás, M. y Palmer, A. (1989). Consumo de drogas en enseñanza media. Comparación entre 1981 y 1988. *Revista Española de Drogodependencias, 14*(1), 9-28.

Calafat, A., Amengual, M., Palmer, A. y Mejias, G. (1994). Modalidades de malestar juvenil y consumo de drogas. *Revista de la Asociación Española de Neuropsiquiatría, 14*(47-48), 65-81.

Calafat, A., Amengual, M., Palmer, A. y Saliba, C. (1997). Drug use and its relationship to other behavior disorders and maladjustment signs among adolescents. *Substance Use & Misuse, 32*(1), 1-24.

Calafat, A., Bohrn, K., Juan, M., Kokkevi, A., Maalste, N., Mendes, F., Palmer, A., Sherlock, K., Simon, J., Stocco, P., Sureda, M.P., Tossmann, P., Van de Wijngaart, G. y Zavatti, P. (1999). *Night life in Europe and recreative drug use. SONAR 98*. Palma de Mallorca: IREFREA and European Commission.

Calafat, A., Juan, M., Becoña, E., Fernández, C., Gil, E., Palmer, A., Sureda, M.P. y Torres, M.A. (2000). *Salir de marcha y consumo de drogas*. Madrid: Plan Nacional de Drogas.

Calafat, A., Stocco, P., Mendes, F., Simon, J., Wijngaart, G., Sureda, M., Palmer, A., Maalsté, N. y Zavatti, P. (1998). *Characteristics and social representation of ecstasy in Europe*. Palma de Mallorca: IREFREA ESPAÑA.

Carpenter, G.A. y Grossberg, S. (1985). Category learning and adaptive pattern recognition, a neural network model. *Proceedings of the Third Army Conference on Applied Mathematics and Computation, ARO Report 86-1*, 37-56.

Carpenter, G.A. y Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing, 37*, 54-115.

Carpenter, G.A. y Grossberg, S. (1987b). ART2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics, 26*, 4919-4930.

Carpenter, G.A. y Grossberg, S. (1990). ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures. *Neural Networks, 3*(4), 129-152.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H. y Rosen, D.B. (1992). Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks, 3*, 698-713.

Carpenter, G.A., Grossberg, S. y Reynolds, J.H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks, 4*, 565-588.

Carpenter, G.A., Grossberg, S. y Rosen, D.B. (1991a). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks, 4*, 493-504.

Carpenter, G.A., Grossberg, S. y Rosen, D.B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4*, 759-771.

Carpintero, H. (1980). La psicología actual desde una perspectiva bibliométrica: Una introducción. *Análisis y Modificación de Conducta, 6*(11-12), 9-23.

Castellanos, J., Pazos, A., Ríos, J. y Zafra, J.L. (1994). Sensitivity analysis on neural networks for meteorological variable forecasting. En J. Vlontzos, J.N. Hwang y E. Wilson (Eds.), *Proceedings of IEEE Workshop on Neural Networks for Signal Processing* (pp. 587-595). New York: IEEE.

Chen, S., Cowan, C.F.N. y Grant, P.M. (1991). Orthogonal least squares learning for radial basis function networks. *IEEE Transactions on Neural Networks, 2*, 302-309.

Cheng, B. y Titterington, D.M. (1994). Neural networks: a review from a statistical perspective. *Statistical Science, 9*(1), 2-54.

Choe, W., Ersoy, O. y Bina, M. (2000). Neural network schemes for detecting rare events in human genomic DNA. *Bioinformatics, 16*(12), 1062-1072.

Clark, G., Hilsenbeck, S., Ravdin, P.M., De Laurentiis, M. y Osborne, C. (1994). Prognostic factors: rationale and methods of analysis and integration. *Breast Cancer Research and Treatment, 32*(1), 105-112.

Cottrell, G.W., Munro, P. y Zipser, D. (1989). Image compression by back propagation: an example of extensional programming. En N.E. Sharkey (Ed.), *Models of cognition: a review of cognitive science* (pp. 208-240). Norwood, NJ: Ablex Publishing Corp.

Cox, D.R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34, 187-202.

Cox, D.R. (1975). Partial likelihood. *Biometrika, 62*, 269-276.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematical Control, Signal and Systems, 2*, 303-314.

De Laurentiis, M. y Ravdin, P.M. (1994a). Survival analysis of censored data: Neural network analysis detection of complex interactions between variables. *Breast Cancer Research and Treatment, 32*, 113-118.

De Laurentiis, M. y Ravdin, P.M. (1994b). A technique for using neural network analysis to perform survival analysis of censored data. *Cancer Letters, 77*, 127-138.

Desieno, D. (1988). Adding a conscience to competitive learning. *Proceedings of the International Conference on Neural Networks, I*, 117-124.

Ebell, M.H. (1993). Artificial neural networks for predicting failure to survive following in-hospital cardiopulmonary resuscitation. *Journal of Family Practice, 36*(3), 297-303.

Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. *Journal of the American Statistical Association, 72*, 557-565.

Elman, J.L. (1990). Finding structure in time. *Cognitive Science, 14*, 179-211.

Engelbrecht, A.P., Cloete, I. y Zurada, J.M. (1995). Determining the significance of input parameters using sensitivity analysis. En J. Mira y F. Sandoval (Eds.), *Proceedings of International Workshop on Artificial Neural Networks* (pp. 382-388). New York: Springer.

Fahlman, S.E. (1988). Faster-learning variations on back-propagation: an empirical study. En D. Touretsky, G.E. Hinton y T.J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 38-51). San Mateo: Morgan Kaufmann.

Fahlman, S.E. y Lebiere, C. (1990). The cascade-correlation learning architecture. En D.S. Touretzky (Ed.), *Advances in neural information processing systems* (pp. 524-532). Los Altos, CA: Morgan Kaufmann Publishers.

Faraggi, D., LeBlanc, M. y Crowley, J. (2001). Understanding neural networks using regression trees: an application to multiple myeloma survival data. *Statistics in Medicine, 20*(19), 2965-2976.

Faraggi, D. y Simon, R. (1995). A neural network model for survival data. *Statistics in Medicine, 14*(1), 73-82.

Fausett, L. (1994). *Fundamentals of neural networks*. New Jersey: Prentice-Hall.

Ferreiro, L. (1993). *Bibliometría (Análisis Bivariante)*. Madrid: EYPASA.

Fiori, S. (2000). An experimental comparison of three PCA neural networks. *Neural Processing Letters, 11*, 209-218.

Fletcher, R. (1987). *Practical methods of optimization*. New York: Wiley.

Flexer, A. (1995). *Connectionist and statisticians, friends or foes ?*. The Austrian Research Institute for Artificial Intelligence. Recuperado 20/01/01, desde acceso FTP: Nombre del servidor: ai.univie.ac.at Archivo: oefai-tr-95-06_ps(1).ps.

Fodor, J.A. y Pylyshyn, Z.W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28*, 3-71.

Fotheringhame, D. y Baddeley, R. (1997). Nonlinear principal components analysis of neuronal spike train data. *Biological Cybernetics, 77*, 283-288.

Frost, F. y Karri, V. (1999). Determining the influence of input parameters on BP neural network output error using sensitivity analysis. En B. Verma, H. Selvaraj, A. Carvalho y X. Yao (Eds.), *Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications* (pp.45-49). Los Alamitos, CA: IEEE Computer Society Press.

Frye, K.E., Izenberg, S.D., Williams, M.D. y Luterman, A. (1996). Simulated biologic intelligence used to predict length of stay and survival of burns. *Journal of Burn Care & Rehabilitation, 17*(6), 540-546.

Fukushima, K. (1975). Cognitron: a self-organizing multilayer neural network. *Biological Cibernetics, 20*, 121-136.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cibernetics, 36*, 193-202.

Fukushima, K. (1988). Neocognitron: a hierarchical neural network model capable of visual pattern recognition. *Neural Networks, 1*(2), 119-130.

Fukushima, K., Miyake, S., e Ito, T. (1983). Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics, 13*, 826-834.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks, 2*, 183-192.

Garrido, L., Gaitan, V., Serra, M. y Calbet, X. (1995). Use of multilayer feedforward neural nets as a display method for multidimensional distributions. *International Journal of Neural Systems, 6*(3), 273-282.

Garson, G.D. (1991). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.

Gedeon, T.D. (1997). Data mining of inputs: analysing magnitude and functional measures. *International Journal of Neural Systems, 8*(2), 209-218.

Gill, P.E., Murray, W. y Wright, M.H. (1981). *Practical optimization*. London: Academic Press.

Glantz, S.A. (1990). *Primer of applied regression and analysis of variance*. New York: McGraw-Hill.

Grimson, W.E.L. y Patil, R.S. (1987). *AI in the 1980s and beyond*. Cambridge, Mass.: The MIT Press.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics, 23*, 121-134.

Grossberg, S. (1982). *Studies of mind and brain: neural principles of learning*. Amsterdam: Reidel Press.

Grözinger, M., Kögel, P. y Röschke, J. (1998). Effects of Lorazepam on the automatic online evaluation of sleep EEG data in healthy volunteers. *Pharmacopsychiatry, 31*(2), 55-59.

Guo, Z. y Uhrig, R.E. (1992). Sensitivity analysis and applications to nuclear power plant. En IEEE (Ed.), *International Joint Conference on Neural Networks* (pp. 453-458). Piscataway, NJ: IEEE.

Hanley, J.A. y McNeil, B.J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*, 29-36.

Hanley, J.A. y McNeil, B.J. (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology, 148*, 839-843.

Hardgrave, B.C., Wilson, R.L. y Walstrom, K.A. (1994). Predicting graduate student success: A comparison of neural networks and traditional techniques. *Computers Operation Research, 21*(3), 249-263.

Harrison, R.F., Marshall, J.M. y Kennedy, R.L. (1991). The early diagnosis of heart attacks: a neurocomputational approach. En IEEE (Ed.), *Proceedings of IEEE International conference on Neural Networks* (pp. 231-239). New York: IEEE.

Hartman, E., Keeler, J.D. y Kowalski, J.M. (1990). Layered neural networks with Gaussian hidden units as universal approximators. *Neural Computation, 2*(2), 210-215.

Hebb, D. (1949). *The organization of behavior*. New York: Wiley.

Hecht-Nielsen, R. (1987). Counterpropagation networks. *Applied Optics, 26,* 4979-4984.

Hecht-Nielsen, R. (1988). Applications of counterpropagation networks. *Neural Networks, 1*, 131-139.

Hecht-Nielsen, R. (1990). *Neurocomputing*. Reading, MA: Addison-Wesley.

Hertz, J., Krogh, A. y Palmer, R. (1991). *Introduction to the theory of neural computation*. Redwood City, CA: Addison-Wesley.

Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence, 40,* 185-234.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, 79*, 2554-2558.

Horgan, J. (1994). Marvin L. Minsky: el genio de la inteligencia artificial. *Investigación y Ciencia, Febrero,* 28-29.

Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

Hosmer, D.W. y Lemeshow, S. (1980). A goodness-of-fit test for the multiple logistic regression model. *Communications in Statistics, A10,* 1043-1069.

Hunter, A., Kennedy, L., Henry, J. y Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine, 62,* 11-19.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*(4), 295-308.

Jordan, M.I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531-546.

Kehoe, S., Lowe, D., Powell, J.E. y Vincente, B. (2000). Artificial neural networks and survival prediction in ovarian carcinoma. *European Journal of Gynaecological Oncology, 21*(6), 583-584.

Kemp, R.A., McAulay, C. y Palcic, B. (1997). Opening the black box: the relationship between neural networks and linear discriminant functions. *Analytical Cellular Pathology, 14*, 19-30.

Kleinbaum, D.G. (1996). *Survival analysis: a self-learning text*. New York: Springer.

Klimasauskas, C.C. (Ed.) (1989). *The 1989 neuro-computing bibliography*. Cambridge: MIT Press.

Klöppel, B. (1994). Neural networks as a new method for EEG analysis: A basic introduction. *Neuropsychobiology, 29*, 33-38.

Kohonen, T. (1977). *Associative memory*. Berlin: Springer-Verlag.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59-69.

Kohonen, T. (1984). *Self-organization and associative memory*. Berlin: Springer.

Kohonen, T. (1988). Learning vector quantization. *Neural Networks, 1*, 303.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer-Verlag.

Kolmogorov, A.N. (1957). On the representation of continuous functions of several variables by means of superpositions of continuous functions of one variable. *Doklady Akademii Nauk SSSR, 114*, 953-956.

Kosko, B. (1992). *Neural networks and fuzzy systems*. Englewood Cliffs, NJ: Prentice-Hall.

Kramer, M.A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChe Journal, 37*(2), 233-243.

Lang, K.J., Waibel, A.H. y Hinton, G. (1990). A time-delay neural network architecture for isolated word recognition. *Neural Networks, 3*, 23-44.

Lapuerta, P., Azen, S.P. y Labree, L. (1995). Use of neural networks in predicting the risk of coronary artery disease. *Computers and Biomedical Research, 28*, 38-52.

Le Cun, Y. (1985). A learning procedure for asymetric threshold network. *Proceedings of Cognitiva, 85*, 599-604.

Liestol, K., Andersen, P. y Andersen, U. (1994). Survival analysis and neural nets. *Statistics in Medicine, 13*(12), 1189-1200.

Lim, T.S., Loh, W.Y. y Shih, Y.S. (1999). *A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms*. Recuperado 10/04/02, desde http://www.recursive-partitioning.com/mach1317.pdf.

Lippmann, R.P. y Shahian, D.M. (1997). Coronary artery bypass risk prediction using neural networks. *Annals of Thoracic Surgery, 63*, 1635-1643.

Lundin, M., Lundin, J., Burke, H.B., Toikkanen, S., Pylkkänen, L. y Joensuu, H. (1999). Artificial neural networks applied to survival prediction in breast cancer. *Oncology, 57*(4), 281-286.

MacWhinney, B. (1998). Models of the emergence of language. *Annual Review of Psychology, 49*, 199-227.

Macy, M. (1996). Natural selection and social learning in prisoner's dilemma: coadaptation with genetic algorithms and artificial neural networks. *Sociological Methods and Research, 25*(1), 103-137.

Marubini, E. y Valsecchi, M.G. (1995). *Analysing survival data from clinical trials and observational studies*. New York: John Wiley and Sons.

Massini, G. y Shabtay, L. (1998). Use of a constraint satisfaction network model for the evaluation of the methadone treatments of drug addicts. *Substance Use & Misuse, 33*(3), 625-656.

Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press.

Maurelli, G. y Di Giulio, M. (1998). Artificial neural networks for the identification of the differences between “light” and “heavy” alcoholics, starting from five nonlinear biological variables. *Substance Use & Misuse, 33*(3), 693-708.

McClelland, J.L., Rumelhart, D.E. y el grupo de investigación PDP (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 2, Psychological and biological models*. Cambridge, Mass.: The MIT Press.

McCullagh, P. y Nelder, J.A. (1989). *Generalized linear models*. London: Chapman & Hall.

McCulloch, W.S. y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115-133.

McCusker, J., Bigelow, C., Frost, R., Garfield, F., Hindin, R., Vickers-Lahti, M. y Lewis, B.F. (1997). The effects of planned duration of residential drug abuse treatment on recovery and HIV risk behavior. *American Journal of Public Health, 87*, 1637-1644.

McCusker, J., Bigelow, C., Vickers-Lahti, M., Spotts, D., Garfield, F. y Frost, R. (1997). Planned duration of residential drug abuse treatment: efficacy versus treatment. *Addiction, 92*, 1467-1478.

McCusker, J., Vickers-Lahti, M., Stoddard, A.M., Hindin, R., Bigelow, C., Garfield, F., Frost, R., Love, C. y Lewis, B.F. (1995). The effectiveness of alternative planned

durations of residential drug abuse treatment. *American Journal of Public Health, 85,* 1426-1429.

Méndez, A. (1986). Los indicadores bibliométricos. *Política Científica, Octubre*, 34-36.

Michie, D., Spiegelhalter, D.J. y Taylor, C.C. (1994). *Machine learning, neural and statistical classification*. New York: Ellis Horwood.

Minsky, M.L. y Papert, S.A. (1969). *Perceptrons: An introduction to computational geometry*. Cambridge, Mass.: The MIT Press.

Minsky, M.L. y Papert, S.A. (1988). *Perceptrons: An introduction to computational geometry* (expanded ed.). Cambridge, Mass.: The MIT Press.

Montaño, J.J. y Palmer, A. (en revisión). Numeric sensitivity analysis applied to feedforward neural networks. *Neural Computing & Applications*.

Montaño, J.J., Palmer, A. y Fernández, C. (2002). Redes neuronales artificiales: abriendo la caja negra. *Metodología de las Ciencias del Comportamiento, 4*(1), 77-93.

Moody, J. y Darken, C.J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation, 1*, 281-294.

Navarro, J.B. y Losilla, J.M. (2000). Análisis de datos faltantes mediante redes neuronales artificiales. *Psicothema, 12*(3), 503-510.

NeuralWare (1995). *Neural Works Professional II Plus ver. 5.2*, NeuralWare Inc., Pittsburgh, PA.

Ogura, H., Agata, H., Xie, M., Odaka, T. y Furutani, H. (1997). A study of learning splice sites of DNA sequences by neural networks. *Biology and Medicine, 27*(1), 67-75.

Ohno-Machado, L. (1996). *Medical applications of artificial neural networks: connectionist models of survival*. Tesis doctoral no publicada. Stanford University.

Ohno-Machado, L. (1997). A comparison of Cox proporcional hazards and artificial neural network models for medical prognosis. *Computational Biology in Medicine, 27*(1), 55-65.

Ohno-Machado, L. y Musen, M.A. (1995). A comparison of two computer-based prognostic systems for AIDS. *Proceedings of the Nineteenth Annual Symposium on Computer Applications in Medical Care,* 737-741.

Ohno-Machado, L. y Musen, M.A. (1997a). Modular neural networks for medical prognosis: quantifying the benefits of combining neural networks for survival prediction. *Connection Science: Journal of Neural Computing, Artificial Intelligence*

and Cognitive Research, 9(1), 71-86.

Ohno-Machado, L. y Musen, M.A. (1997b). Sequential versus standard neural networks for pattern recognition: an example using the domain of coronary heart disease. *Computational Biology in Medicine, 27*(4), 267-281.

Ohno-Machado, L., Walker, M. y Musen, M.A. (1995). Hierarchical neural networks for survival analysis. *Medinfo, 8 Pt 1*, 828-832.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology, 15*, 267-273.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems, 1*, 61-68.

Olazarán, M. (1991). *A historical sociology of neural network research*. Tesis doctoral no publicada. University of Edinburgh.

Olazarán, M. (1993). Controversias y emergencia del conexionismo: Una perspectiva histórica y sociológica. *Revista Internacional de Sociología, 4*, 91-122.

Olson, S.J. y Grossberg, S. (1998). A neural network model for the development of simple and complex cell receptive fields within cortical maps of orientation and ocular dominance. *Neural Networks, 11*(2), 189-208.

Palmer, A. (1985). *Análisis de la supervivencia*. Barcelona: Ed. Universitat Autónoma de Barcelona.

Palmer, A. (1993a). M-estimadores de localización como descriptores de las variables de consumo. *Adicciones, 5*(2), 171-184.

Palmer, A. (1993b). Modelo de regresión de Cox: ejemplo numérico del proceso de estimación de parámetros. *Psicothema,5*(2), 387-402.

Palmer, A., Amengual, M. y Calafat, A. (1992). ¿Cuánto alcohol consumen realmente los jovenes?: Una técnica de análisis. *Adicciones, 5*(1), 75-78.

Palmer, A. y Cajal, B. (1996). El estudio del cambio desde la perspectiva del Análisis Histórico del Cambio. *Revista de Psicología General y Aplicada, 49*(1), 83-102.

Palmer, A. y Losilla, J.M. (1998). El análisis de la supervivencia. En J. Renom (Coord.), *Tratamiento informatizado de datos* (pp. 193-227). Barcelona: Editorial Masson S.A.

Palmer, A. y Losilla, J.M. (1999, Septiembre). *El análisis de datos de supervivencia mediante redes neuronales artificiales*. Trabajo presentado en el VI Congreso de Metodología de las Ciencias Sociales y de la Salud, Oviedo (Spain).

Palmer, A. y Montaño, J.J. (1999). ¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones. *Adicciones, 11*(3), 243-255.

Palmer, A. y Montaño, J.J. (2002). Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo. *Psicothema, 14*(3), 630-636.

Palmer, A., Montaño, J.J. y Calafat, A. (2000). Predicción del consumo de éxtasis a partir de redes neuronales artificiales. *Adicciones, 12*(1), 29-41.

Palmer, A., Montaño, J.J. y Fernández, C. (en revisión). Sensitivity Neural Network: an artificial neural network simulator with sensitivity analysis. *Behavior Research: Methods, Instruments, & Computers*.

Palmer, A., Montaño, J.J. y Jiménez, R. (2001). Tutorial sobre redes neuronales artificiales: el perceptrón multicapa. *Psicologia.com* (Revista electrónica), *5*(2). Dirección URL: http://www.psicologia.com.

Palmer, A., Montaño, J.J. y Jiménez, R. (2002). Tutorial sobre redes neuronales artificiales: los mapas autoorganizados de Kohonen. *Psicologia.com* (Revista electrónica), *6*(1). Dirección URL: http://www.psicologia.com.

Parker, D. (1985). *Learning logic* (Informe técnico Nº TR-87). Cambridge: Center for Computational Research in Economics and Management Science.

Parmar, M.K.B. y Machin, D. (1995). *Survival analysis: a practical approach*. New York: John Wiley and Sons.

Penny, W.D. y Frost, D.P. (1997). Neural network modeling of the level of observation decision in an acute psychiatric ward. *Computers and Biomedical Research, 30*, 1-17.

Pineda, F.J. (1989). Recurrent back-propagation and the dynamical approach to neural computation. *Neural Computation, 1*, 161-172.

Pitarque, A., Roy, J.F. y Ruíz, J.C. (1998). Redes neurales vs. modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación. *Psicológica, 19*, 387-400.

Pitarque, A., Ruíz, J.C., Fuentes, I., Martínez, M.J. y García-Merita, M. (1997). Diagnóstico clínico en psicología a través de redes neurales. *Psicothema, 9*(2), 359-363.

Plan Nacional sobre Drogas (2000). *Memoria 2000*. Madrid: Plan Nacional sobre Drogas.

Prentice, R.L. y Kalbfleisch, J.D. (1979). Hazard rate models with covariates. *Biometrics, 35*, 25-39.

Rambhia, A.H., Glenny, R. y Hwang, J. (1999). Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis. En IEEE

(Ed.), *IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 1097-1100). Piscataway, NJ: IEEE.

Ravdin, P.M. y Clark, G.M. (1992). A practical application of neural network analysis for predicting outcome of individual breast cancer patients. *Breast Cancer Research and Treatment, 22*(3), 285-293.

Reason, R. (1998). How relevant is connectionist modelling of reading to educational practice? Some implications of Margaret Snowling's article. *Educational and Child Psychology, 15*(2), 59-65.

Riedmiller, M. y Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *IEEE International Conference on Neural Networks,* 586-591.

Ripley, B.D. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society, 56*(3), 409-456.

Rojas, R. (1996). *Neural networks: a systematic introduction*. Berlin: Springer.

Romera, M.J. (1992). Potencialidad de la bibliometría para el estudio de la ciencia. Aplicación a la educación especial. *Revista de Educación, 297,* 459-478.

Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65,* 386-408.

Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En D.E. Rumelhart y J.L. McClelland (Eds.), *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rumelhart, D.E., McClelland, J.L. y el grupo de investigación PDP (1986). *Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, Foundations*. Cambridge, Mass.: The MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet*. New York: Springer-Verlag.

Sancho, R. (1990). Indicadores bibliométricos utilizados en la evaluación de la ciencia y la tecnología. Revisión bibliográfica. *Revista Española de Documentación Científica, 13*(3-4), 842-865.

Sanger, T.D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks, 2,* 459-473.

Sarle, W.S. (1994). Neural networks and statistical models. En SAS Institute (Ed.), *Proceedings of the 19th Annual SAS Users Group International Conference* (pp.1538-1550). Cary, NC: SAS Institute.

Sarle, W.S. (2000). *How to measure importance of inputs?* Recuperado 2/11/01, desde ftp://ftp.sas.com/pub/neural/importance.html.

Sarle, W.S. (Ed.) (2002). *Neural network FAQ*. Recuperado 20/04/02, desde ftp://ftp.sas.com/pub/neural/FAQ.html.

Sesé, A.., Palmer, A., Montaño, J.J., Jiménez, R., Sospedra, M.J. y Cajal, B. (2001, Septiembre). *Redes neuronales artificiales y técnicas clásicas de reducción de la dimensionalidad en modelos psicométricos de medida: un estudio comparativo*. Trabajo presentado en el VII Congreso de Metodología de las Ciencias Sociales y de la Salud, Madrid (Spain).

Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.

Somoza, E. y Somoza, J.R. (1993). A neural-network approach to predicting admission decisions in a psychiatric emergency room. *Medicine Decision Making, 13*(4), 273-280.

Spackman, K.A. (1992). Maximum likelihood training of connectionist models: comparison with least-squares backpropagation and logistic regression. En IEEE (Ed.), *Proceedings of the 15th Annual Symposium of Computer Applications in Medical Care* (pp. 285-289). New York: Institute of Electrical and Electronics Engineers.

Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks, 3*, 110-118.

Specht, D.F. (1991). A generalized regression neural network. *IEEE Transactions on Neural Networks, 2*, 568-576.

Speight, P.M., Elliott, A.E., Jullien, J.A., Downer, M.C. y Zakzrewska, J.M. (1995). The use of artificial intelligence to identify people at risk of oral cancer and precancer. *British Dental Journal, 179*(10), 382-387.

Speri, L., Schilirò, G., Bezzetto, A., Cifelli, G., De Battisti, L., Marchi, S., Modenese, M., Varalta, F. y Consigliere, F. (1998). The use of artificial neural networks methodology in the assessment of “vulnerability” to heroin use among army corps soldiers: A preliminary study of 170 cases inside the Military Hospital of Legal Medicine of Verona. *Substance Use & Misuse, 33*(3), 555-586.

Swets, J.A. (1973). The relative operating characteristic in psychology. *Science, 182*,

990-1000.

Swets, J.A. (1986). Form of empirical ROCs in discrimination and diagnostic tasks: Implications for theory and measurement of performance. *Psychological Bulletin, 99*, 181-198.

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*, 1285-1293.

Takenaga, H., Abe, S., Takatoo, M., Kayama, M., Kitamura, T. y Okuyama, Y. (1991). Input layer optimization of neural networks by sensitivity analysis and its application to recognition of numerals. *Transactions of the Institute of Electrical Engineers Japan, 111*(1), 36-44.

Tchaban, T., Taylor, M.J. y Griffin, A. (1998). Establishing impacts of the inputs in a feedforward network. *Neural Computing & Applications, 7*, 309-317.

Thrun, S., Mitchell, T. y Cheng, J. (1991). The MONK's comparison of learning algorithms: introduction and survey. En S. Thrun, J. Bala, E. Bloedorn y I. Bratko (Eds.), *The MONK's problem: a performance comparison of different learning algorithms* (pp. 1-6). Pittsburg: Carnegie Mellon University.

Tsiatis, A. (1981). A large sample study of Cox's regression model. *Annals of Statistics, 9*, 93-108.

Tsui, F.C. (1996). *Time series prediction using a multiresolution dynamic predictor: Neural network*. Tesis doctoral no publicada. University of Pittsburgh.

Turner, D.A. (1978). An intuitive approach to receiver operating characteristic curve analysis. *The Journal of Nuclear Medicine, 19*, 213-220.

Van Ooyen, A. y Nienhuis, B. (1992). Improving the convergence of the backpropagation algorithm. *Neural Networks, 5*, 465-471.

Vicino, F. (1998). Some reflections on artificial neural networks and statistics: two ways of obtaining solutions by working with data. *Substance Use & Misuse, 33*(2), 221-231.

Von Der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striata cortex. *Kybernetik, 14*, 85-100.

Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psychometrika, 63*(1), 5-22.

Wan, E.A. (1990). Temporal backpropagation: an efficient algorithm for finite impulse response neural networks. En D.S. Touretzky, J.L. Elman, T.J. Sejnowski y G.E. Hinton (Eds.), *Proceedings of the 1990 Connectionist Models Summer School* (pp. 131-140). San Mateo, CA: Morgan Kaufmann.

Wang, W., Jones, P. y Partridge, D. (2000). Assessing the impact of input features in a feedforward neural network. *Neural Computing & Applications, 9*, 101-112.

Weinstein, M.C. y Fineberg, H.V. (1980). *Clinical decision analysis*. Philadelphia: W.B. Saunders Company.

Werbos, P.J. (1974). *Beyond regression: new tools for prediction an analysis in behavioral sciences*. Tesis doctoral no publicada. Harvard University.

Werbos, P.J. (1990). Backpropagation through time: What it is and how to do it. *Proceedings of the IEEE, 78*, 1550-1560.

White, H. (1989). Neural network learning and statistics. *AI Expert*, December, 48-52.

Widrow, B. y Hoff, M. (1960). Adaptive switching circuits. En J. Anderson y E. Rosenfeld (Eds.), *Neurocomputing* (pp. 126-134). Cambridge, Mass.: The MIT Press.

Williams, R.J. y Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation, 1*, 270-280.

Williamson, J.R. (1995). *Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps* (Informe técnico Nº CAS/CNS-95-003). Boston: Boston University, Center of Adaptive Systems and Department of Cognitive and Neural Systems.

Zou, Y., Shen, Y., Shu, L., Wang, Y., Feng, F., Xu, K., Qu, Y., Song, Y., Zhong, Y., Wang, M. y Liu, W. (1996). Artificial neural network to assist psychiatric diagnosis. *British Journal of Psychiatry, 169*, 64-67.

2. Publicaciones

2.1.
¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones.

¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones

Palmer Pol, A*; Montaño Moreno, J.J.**

* Prof. Titular del Departamento de Psicología. Universidad de las Islas Baleares.
** Becario. Departamento de Psicología. Universidad de las Islas Baleares.

Enviar correspondencia a:
Alfonso Palmer Pol. Universidad de las Islas Baleares. Departamento de Psicología. Cra. de Valldemossa, km. 7,5. 07071 Palma (Baleares). Teléfono 971173432.

Resumen:

En el presente trabajo, se introduce al lector en el campo de las redes neuronales artificiales (RNA) –características generales, arquitecturas, reglas de aprendizaje, ejemplos ilustrativos y aplicaciones generales—, y se realiza una revisión de las aplicaciones llevadas a cabo con esta tecnología en el campo de las conductas adictivas. Los resultados de las investigaciones demuestran la capacidad de las RNA para predecir el consumo de drogas, extraer las características prototípicas del sujeto adicto y seleccionar el tratamiento más adecuado en función de esas características. Aunque tales estudios son preliminares, los resultados se pueden considerar prometedores, perfilándose las RNA como un potente instrumento al servicio del profesional dedicado al campo de las conductas adictivas.

Palabras clave: *redes neuronales artificiales; adicción a las drogas; predicción; revisión bibliográfica.*

Abstract:

In this paper, we introduce to the reader in the field of artificial neural networks (ANN) –general features, architectures, learning rules, illustrative examples and general applications—, and we review the applications carried out with this technology in the field of addictive behaviors. Results of research show the capacity of ANN in order to predict drug consumption, extract prototype characteristics of addicted subjects and choose the treatment most appropiate according to those characteristics. Although these studies are preliminary, the results can be qualified as very promising; so, ANN are a powerful tool for professional dedicated to field of addictive behaviors.

Key words: *artificial neural networks, drug addiction, prediction, bibliographic review.*

INTRODUCCIÓN

El uso y abuso de sustancias comprende un conjunto de conductas complejas que son iniciadas, mantenidas y modificadas por una variedad de factores conocidos y desconocidos. El tipo de función o relación que se establece entre la conducta adictiva y los factores que la explican no se puede reducir a una simple relación lineal de "causa-efecto" (Buscema, 1997, 1998). Por tanto, si nos planteamos como objetivo la prevención y la predicción de este tipo de conductas, será necesario utilizar instrumentos capaces de manejar relaciones complejas o no lineales.

El reciente campo de la computación biológica —que comprende las redes neuronales artificiales, los algoritmos genéticos, las estrategias y programación evolutivas, los sistemas borrosos y la vida artificial (Pazos, 1996)—, en general, y las redes neuronales artificiales (RNA) en particular, han demostrado su utilidad en la solución de problemas complejos. Así, las RNA han sido utilizadas satisfactoriamente en la predicción de diversos problemas en diferentes áreas de conocimiento —biología, medicina, economía, ingeniería y psicología— (Arbib, 1995; Simpson, 1995; Arbib, Erdi y Szentagothai, 1997); con buenos resultados respecto a los modelos derivados de la estadística clásica (Bonilla y Puertas, 1997; Duncan, 1997; French, Dawson y Dobbs, 1997; Jefferson, Pendleton, Lucas et al., 1997; Shekharan, 1997; Tommaso, Sciruicchio, Bellotti et al., 1997; Vohradsky, 1997; West, Brockett y Golden, 1997; De Lillo y Meraviglia, 1998; Jang, 1998; Waller, Kaiser, Illian et al., 1998). En el caso de las adicciones, estudios recientes, que se describen más adelante, demuestran la capacidad de las RNA para predecir el consumo de drogas, extraer las características prototípicas del sujeto adicto y

seleccionar el tratamiento más adecuado en función de esas características.

Con el presente trabajo nos proponemos introducir al lector, de una forma sencilla, en el campo de las RNA y revisar las aplicaciones llevadas a cabo con esta tecnología en el ámbito del estudio de las adicciones.

REDES NEURONALES ARTIFICIALES

Características generales

Las RNA son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas (Hilera y Martínez, 1995). Consisten en un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. Los pesos representan la información que será usada por la red neuronal para resolver un problema determinado.

Así, las RNA son sistemas adaptativos que aprenden de la experiencia, esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos.

Mediante este entrenamiento o aprendizaje, las RNA crean su propia representación interna del problema, por tal motivo se dice que son autoorganizadas. Posteriormente, pueden responder adecuadamente cuando se les presentan situaciones a las que no habían sido expuestas anteriormente, es decir, las RNA son capaces de generalizar de casos anteriores a casos nuevos.

Esta característica es fundamental ya que permite a la red responder correctamente no sólo ante informaciones novedosas, sino también ante informaciones distorsionadas o incompletas.

En las RNA el tipo de procesamiento de la información es en paralelo, en el sentido de que muchas neuronas pueden estar funcionando al mismo tiempo. De hecho, nuestro cerebro está compuesto por unas 10^{11} neuronas, las cuales operan en paralelo. Es ahí donde reside una parte fundamental de su poder de procesamiento. Aunque individualmente las neuronas sean capaces de realizar procesamientos muy simples, ampliamente interconectadas a través de las sinapsis (cada neurona puede conectarse con otras 10.000 en promedio) y trabajando en paralelo pueden desarrollar una actividad global de procesamiento impresionante.

Biológicamente, se suele aceptar que el conocimiento está más relacionado con las conexiones entre neuronas que con las propias neuronas (Alkon, 1989; Shepherd, 1990); es decir, el conocimiento se encuentra distribuido por las sinapsis de la red. Este tipo de representación distribuida del conocimiento implica que si una sinapsis resulta dañada, no perdemos más que una parte muy pequeña de la información. Además, los sistemas neuronales biológicos son redundantes, de modo que muchas neuronas y sinapsis pueden realizar un papel similar; en definitiva, el sistema resulta tolerante a fallos. En este sentido, sabemos que cada día mueren miles de neuronas en nuestro cerebro, y sin embargo tienen que pasar muchos años para que se resientan nuestras capacidades. De forma análoga, en el caso de las RNA se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones entre neuronas.

El tipo de representación de la información que manejan las RNA tanto en los pesos de las conexiones como en las entradas y salidas de información es numérica. Por ejemplo, un dato de entrada puede consistir en un valor real continuo como la edad de una persona o puede consistir en un valor numérico discreto o binario como el sexo de una persona codificado, por ejemplo, mediante: 0 = hombre, 1 = mujer.

En síntesis, podemos decir que las RNA se inspiran en la estructura del sistema nervioso, con la intención de construir sistemas de procesamiento de la información paralelos, distribuidos y adaptativos que pueden presentar un cierto comportamiento inteligente (Martín del Brío y Sanz, 1997).

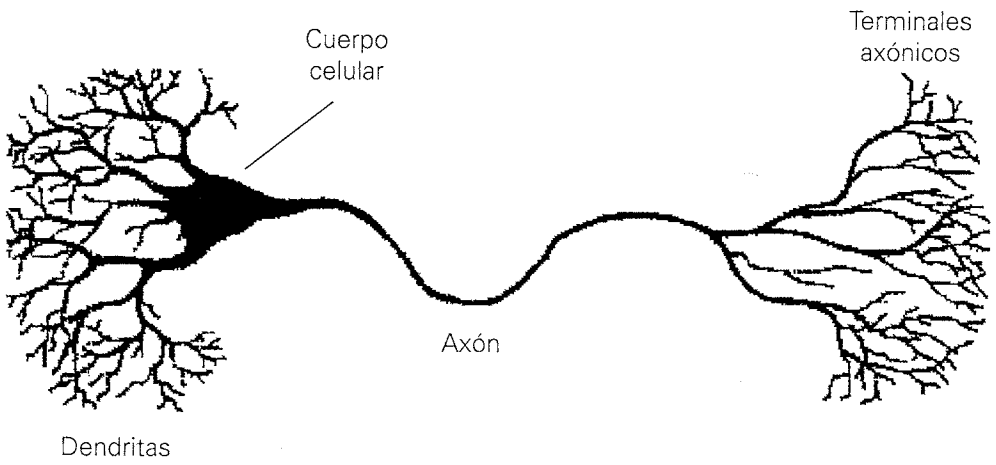
Estas características contrastan con la estructura y funcionamiento de un ordenador convencional. Este tipo de computadores son máquinas construidas en torno a un único procesador (hardware) que ejecuta de un modo secuencial (paso a paso) un programa (software) almacenado en su memoria. Siguiendo este esquema, los ordenadores convencionales pueden realizar importantes operaciones de cálculo y razonamiento lógico, de forma mucho más rápida y eficiente que el cerebro. Sin embargo, existen problemas de difícil solución para un ordenador convencional que el cerebro resuelve eficazmente (Hertz, Krogh y Palmer, 1991). Precisamente estos problemas son los relacionados con el mundo real, los cuales están caracterizados por un alto grado de complejidad, imprecisión e incertidumbre como es el caso de la toma de decisiones, el reconocimiento de patrones como el habla, imágenes o caracteres escritos, etc..

La neurona artificial

Las neuronas biológicas (figura 1) se caracterizan por su capacidad de comunicarse. Las dendritas y el cuerpo celular de la neurona reciben señales de entrada excitatorias e inhibitorias de las neuronas vecinas; el cuerpo celular las combina e integra y emite seña-

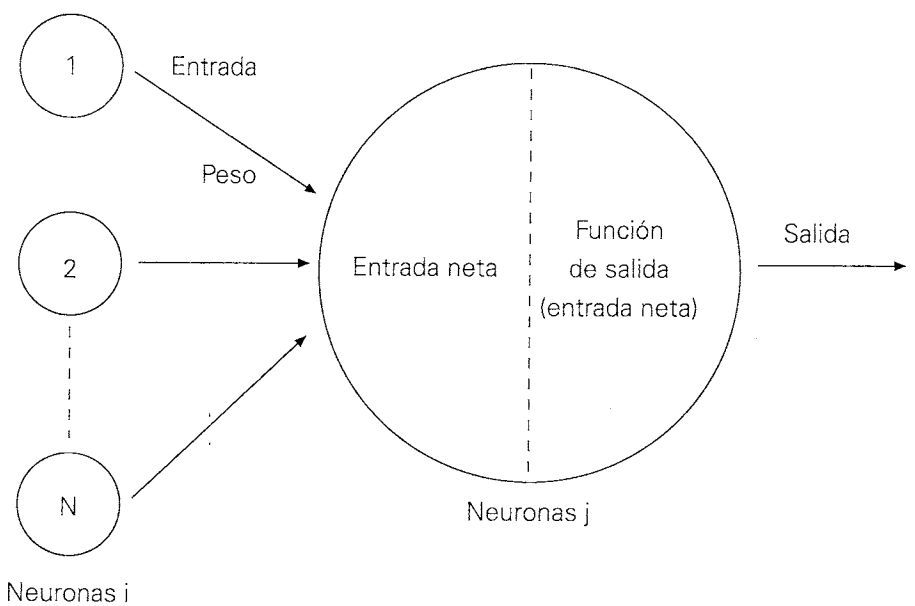
les de salida. El axón transporta esas señales a los terminales axónicos, que se encargan de distribuir información a un nuevo conjunto de neuronas. Por lo general, una neurona recibe información de miles de otras neuronas y, a su vez, envía información a miles de neuronas más.

Figura 1. Estructura general de una neurona biológica.



Por su parte, la neurona artificial pretende mimetizar las características más importantes de la neurona biológica. En general, recibe las señales de entrada de las neuronas vecinas ponderadas por los pesos de las conexiones. La suma de estas señales ponderadas proporciona la entrada total o neta de la neurona y, mediante la aplicación de una función matemática —denominada función de salida—, sobre la entrada neta, se calcula un valor de salida, el cual es enviado a otras neuronas (figura 2). Tanto los valores de entrada a la neurona como su salida pueden ser señales excitatorias (cuando el valor es positivo) o inhibitorias (cuando el valor es negativo).

Figura 2. Funcionamiento general de una neurona artificial.



Arquitecturas

Las neuronas que componen una RNA se organizan de forma jerárquica formando capas. Una capa o nivel es un conjunto de neuronas cuyas entradas de información provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas de información se dirigen al mismo destino (que puede ser otra capa de neuronas). En este sentido, se distinguen tres tipos de capas: la capa de entrada recibe la información del exterior; la o las capas ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema y, por tanto, no tienen contacto con el exterior; por último, la capa de salida envía la respuesta de la red al exterior.

En función de la organización de las neuronas en la red formando capas o agrupaciones podemos encontrarnos con dos tipos de arquitecturas básicas: redes multicapa y redes monocapa.

Las redes multicapa disponen de conjuntos de neuronas agrupadas en dos o más capas. En la mayoría de casos, este tipo de redes están formadas por una capa de entrada, una capa de salida y una o más capas intermedias u ocultas; donde la información se transmite desde la capa de entrada hasta la capa de salida y donde cada neurona está conectada con todas las neuronas de la siguiente capa (en la figura 4 se muestra un ejemplo de red multicapa). Las redes multicapa se suelen utilizar en tareas denominadas heteroasociativas. De lo que se trata es que la red aprenda parejas de datos, de forma que cuando se presenta cierta información de entrada A, deberá responder generando la correspondiente salida asociada B. Por tal motivo, las redes que llevan a cabo este tipo de tareas también reciben el nombre de redes heteroasociativas ya que intentan asociar pares de informaciones distintas. Este tipo de redes son útiles para la clasificación de patrones –ya que, en este caso, se asocia el ejemplo con la clase o categoría a la que pertenece—, y la aproximación de funciones –donde se asocia una información de entrada con otra información de salida.

El tipo de arquitectura multicapa descrito se denomina perceptrón multicapa y ha sido el más ampliamente utilizado en el campo aplicado. La utilidad del perceptrón multicapa reside en su habilidad para operar como aproximador universal de funciones, es decir, este tipo de redes pueden aprender virtualmente cualquier relación entre un conjunto de variables de entrada y salida. Esta habilidad es el resultado de la adopción, por parte de las neuronas de la capa oculta, de una función de salida no lineal (Rumelhart y McClelland, 1986; Masters, 1993; Smith, 1993; Rzempoluck, 1998). Por su parte, el análisis discriminante lineal derivado de la estadística clásica no posee la capacidad de calcular funciones no lineales y, por tanto, presentará un rendimiento inferior frente al perceptrón multicapa en tareas de clasificación que impliquen relaciones no lineales complejas.

Por su parte, las redes monocapa están organizadas, como el propio nombre indica, en una sola capa de neuronas (en la figura 5 se muestra un ejemplo de red monocapa). Cada neurona está conectada con todas las demás que forman la arquitectura. Este tipo de redes se suelen utilizar en tareas denominadas autoasociativas. Para ello, se almacena en los pesos de la red ciertas informaciones mediante una etapa de entrenamiento. Posteriormente, cuando se presenta una información a la entrada de la red, ésta responde proporcionando la información más parecida de las almacenadas. Por tal motivo, las redes que llevan a cabo este tipo de tareas también reciben el nombre de redes autoasociativas ya que intentan asociar una información consigo misma. Este tipo de redes son útiles para regenerar informaciones de entrada, por ejemplo imágenes, que se presentan a la red incompletas o distorsionadas.

Aprendizaje

Como hemos visto, el conocimiento de una RNA se encuentra distribuido en los pesos de las conexiones entre las neuronas que forman la red. Todo proceso de aprendizaje implica cierto número de cambios en estas conexiones. En realidad, puede decirse que se aprende modificando los valores de los pesos de la red en respuesta a un conjunto de ejemplos denominado grupo de entrenamiento. Actualmente existen muchos criterios para modificar los pesos de la red y así conseguir que aprenda a solucionar un determinado problema; estos criterios se denominan, de forma genérica, reglas de aprendizaje. Las reglas de aprendizaje consisten generalmente en algoritmos matemáticos que pueden llegar a ser sumamente complejos. Se suelen considerar dos tipos de reglas de aprendizaje: aprendizaje supervisado y aprendizaje no supervisado.

En el aprendizaje supervisado hay un "profesor" o supervisor que controla el proceso de aprendizaje de la red. El supervisor comprueba la salida de la red en respuesta a una determinada entrada y en el caso de que la salida no coincida con la deseada, se procede a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada. Este tipo de aprendizaje es muy útil para la clasificación de patrones y para la aproximación de funciones.

Con el aprendizaje no supervisado también denominado autoorganizado, la red no requiere influencia de un "profesor" para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la

salida generada en respuesta a una determinada entrada es o no correcta. Su función consiste en encontrar las características, regularidades o categorías que se puedan establecer entre los datos que se presentan en su entrada. Este tipo de aprendizaje se suele utilizar en tareas autoasociativas y en la agrupación de datos en función de su similitud.

Una vez obtenidos y guardados los pesos óptimos en la fase de entrenamiento, debemos medir la eficacia de la red de forma objetiva mediante la presentación de casos nuevos (diferentes a los casos de entrenamiento), de forma que a la fase de entrenamiento le debe seguir una fase de test. En esta fase no se modifican los pesos, simplemente se presentan casos nuevos –llamados casos de test—, a la entrada de la red y ésta proporciona una salida para cada uno de ellos. Si se comprueba que se siguen obteniendo resultados dentro del margen de error deseado, se puede proceder a emplear la RNA dentro de su entorno de trabajo real.

En la metodología de las RNA, con el fin de encontrar la red que tiene la mejor ejecución con casos nuevos –es decir, que sea capaz de generalizar—, la muestra de datos es a menudo subdividida en tres grupos (Masters, 1993; Bishop, 1995; Ripley, 1996; Martín del Brío y Sanz, 1997; Sarle, 1998): entrenamiento, validación y test.

Durante el proceso de entrenamiento o aprendizaje de una red neuronal supervisada, del tipo perceptrón multicapa, los pesos son modificados de forma iterativa de acuerdo con los valores del grupo de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red y la salida deseada por el usuario. De forma característica, en las primeras fases del aprendizaje la red se va adaptando progresivamente al conjunto de datos de entrenamiento, acomodándose al problema y favoreciendo la generalización. Así, se puede observar que el error que comete la red ante los datos de entrenamiento va descendiendo paulatinamente hasta alcanzar un valor mínimo. Sin embargo, a partir de un momento dado el sistema puede comenzar a ajustarse demasiado a las particularidades irrelevantes (ruido) presentes en los patrones de entrenamiento en vez de ajustarse a la función subyacente que relaciona entradas y salidas. Llegados a este punto se dice que la red ha sufrido un sobreentrenamiento o sobreaprendizaje, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos.

Con el fin de evitar el problema del sobreentrenamiento, que puede darse en las redes del tipo perceptrón multicapa, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, denominado grupo de validación, que permita controlar el proceso de aprendizaje. De este modo, a lo largo del aprendizaje la red va modificando los pesos en función de los datos de entrenamiento y de forma alternada se va obteniendo el error que comete la red ante los datos de validación. Esto permite estimar el error de generalización de la red –es decir, el error que se comete ante patrones diferentes a los utilizados en el entrenamiento—, a partir del error que comete ante los patrones de validación (error de validación) a lo largo del proceso de aprendizaje. Normalmente, en las primeras fases del entrenamiento el error de validación va disminuyendo progresivamente hasta un punto a partir del cual este error comienza a aumentar, ese punto indica que la red empieza a aprender las particularidades del grupo de entrenamiento –se produce el sobreentrenamiento. Una práctica común, con el fin de evitar el sobreentrenamiento, consiste en detener el aprendizaje cuando el error de validación alcanza el punto mínimo.

La utilización de un grupo de validación también permite afinar los parámetros de la red, por ejemplo, para seleccionar el número óptimo de unidades ocultas. Así, la arquitectura que obtenga el menor error de validación será la seleccionada.

El error que se obtiene ante los datos de validación proporciona una estimación sesgada del error de generalización de la red seleccionada ya que, aunque indirectamente, el grupo de validación ha intervenido en el entrenamiento. Por tanto, si se desea medir de una forma completamente objetiva la eficacia final del sistema construido, se debe contar con un tercer grupo de datos independiente, denominado grupo de test. El error que comete la red entrenada ante los datos de test proporciona una estimación insesgada del error de generalización. Finalmente, si se comprueba que se siguen obteniendo resultados satisfactorios con el grupo de test, se puede proceder a emplear el modelo dentro de su entorno de trabajo real.

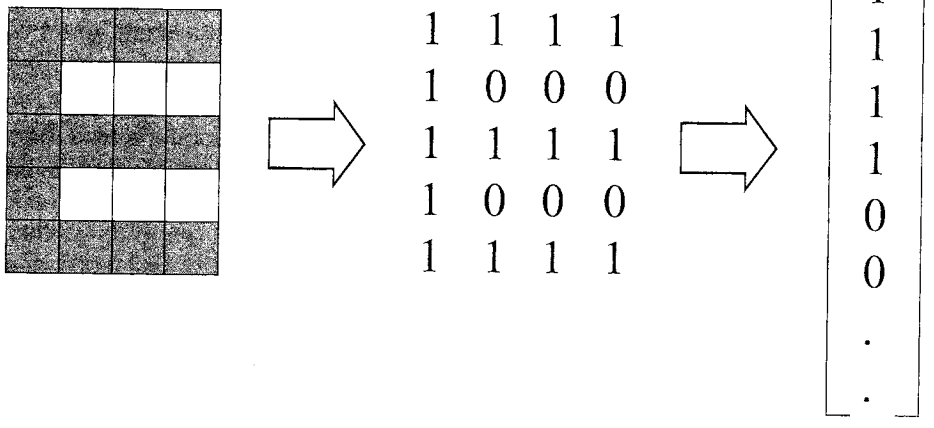
Un ejemplo

Con el fin de ilustrar el proceso de entrenamiento y de test de una red, a continuación expondremos un ejemplo sencillo de reconocimiento de patrones. Más concretamente, se pretende que la red neuronal aprenda a reconocer las figuras de las cinco vocales. Diseñaremos dos tipos de redes neuronales, una llevará a cabo una tarea heteroasociativa, la otra realizará una tarea autoasociativa.

Debido a que las RNA son sistemas adaptativos que aprenden a partir de ejemplos, éstos deben ser representativos de lo que sucede en la realidad. En nuestro caso, tendremos que diseñar diferentes tipos de figuras para cada vocal en función de parámetros tales como el tamaño, posición y estilo de letra. Esto facilitará la capacidad de generalización de la red para responder ante patrones diferentes a los utilizados en el entrenamiento.

La primera tarea que debemos realizar, una vez diseñados los ejemplares o patrones, consiste en tratar las informaciones de forma que la red pueda procesarlas. Recordemos que el tipo de información que maneja una RNA es de tipo numérica: continua o discreta. Pues bien, en el ejemplo que nos ocupa podemos representar las figuras de las vocales mediante un cierto número de píxeles (contracción de los vocablos picture elements, o elementos de imagen). Con el fin de simplificar el problema, imaginemos que queremos representar cada figura mediante una matriz de 5x4 píxeles. Los píxeles negros pueden representarse mediante el valor binario 1 y los blancos con el valor 0. Para cada figura obtendremos un vector de 1s y 0s formado a partir de la configuración de los 20 píxeles resultantes. En la figura 3 se muestra el proceso de codificación de un ejemplar de la vocal E.

Figura 3. Codificación de un ejemplar de la vocal E.



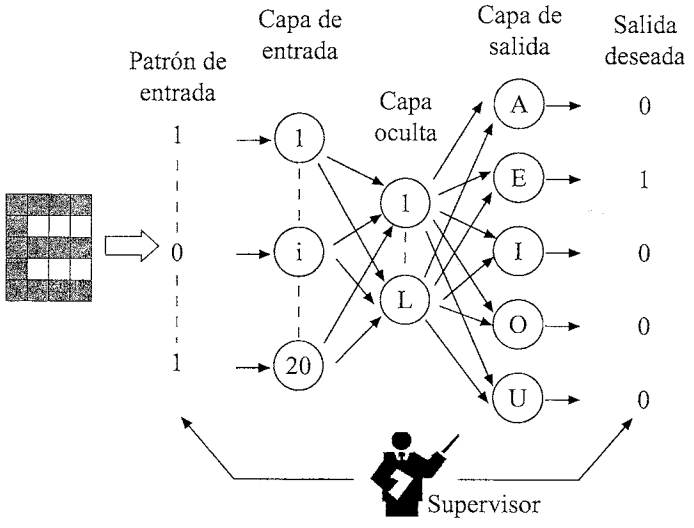
Podemos estar interesados en entrenar la red para clasificar cada figura en la categoría a la que pertenece. En este caso la red debe aprender a asociar cada figura con la vocal que representa (heteroasociación). El tipo de arquitectura que se suele utilizar en este tipo de problemas consiste en un perceptrón multicapa compuesto por una capa de entrada, una oculta y una de salida. El número de neuronas de entrada y de salida estará determinado por el problema. Así, la capa de entrada a la red estará formada por tantas neuronas de entrada como elementos o píxeles formen las figuras; en este caso tenemos 20 píxeles. Cada una de estas neuronas de entrada se encargará de recibir y procesar un píxel. La capa de salida estará formada por tantas neuronas como categorías o clases contenga el problema; en este caso tenemos cinco vocales. Cada neurona de salida representará una vocal. Podemos determinar la salida de la red de forma que ante la presentación de un ejemplar, la neurona de salida correspondiente a la vocal que representa el ejemplar, dé como salida el valor 1 (activada) y todas las demás den como salida el valor 0 (desactivada). Así, si la figura que presentamos a la entrada de la red es una A, entonces la salida de la red debería ser el vector 1 0 0 0 0; si el ejemplar es una E, entonces la salida debería ser el vector 0 1 0 0 0; y así sucesivamente. Por último, el número de neuronas ocultas dependerá, en gran medida, de la complejidad del problema.

La fase de entrenamiento o aprendizaje consistirá en la presentación repetida de un grupo representativo de ejemplos de vocales junto con sus salidas correspondientes. La regla de aprendizaje será supervisada, debido a que cada información de entrada está asociada a una salida deseada. Mediante esta regla iremos modificando los pesos de las conexiones iterativamente hasta que la salida de la red coincida o se aproxime hasta un nivel aceptable a la salida deseada para cada uno de los ejemplos de entrenamiento. En la figura 4A se muestra este proceso para el caso de un ejemplar de la vocal E. En esta fase, la red organiza una representación interna del conocimiento en los pesos de las conexiones de las neuronas ocultas, a fin de aprender la relación que existe entre el conjunto de patrones dados como ejemplo y sus salidas correspondientes.

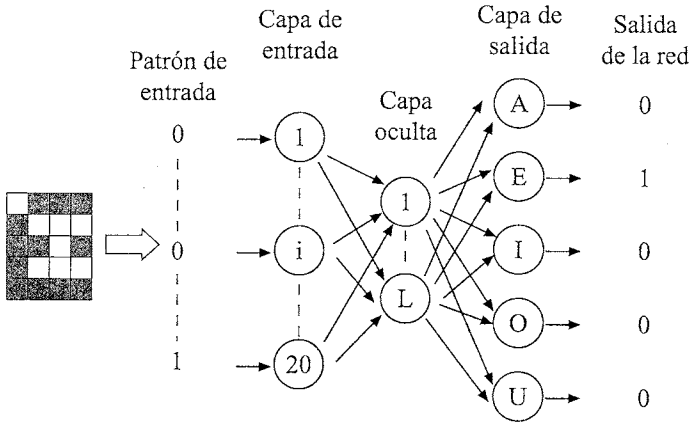
En la fase de test podremos presentar ejemplares nuevos, la red propagará la información a través de las sucesivas capas hasta proporcionar una salida. La pre-

Figura 4. Entrenamiento y test de un perceptrón multicapa supervisado para la clasificación de las vocales.

A) Fase de entreno:



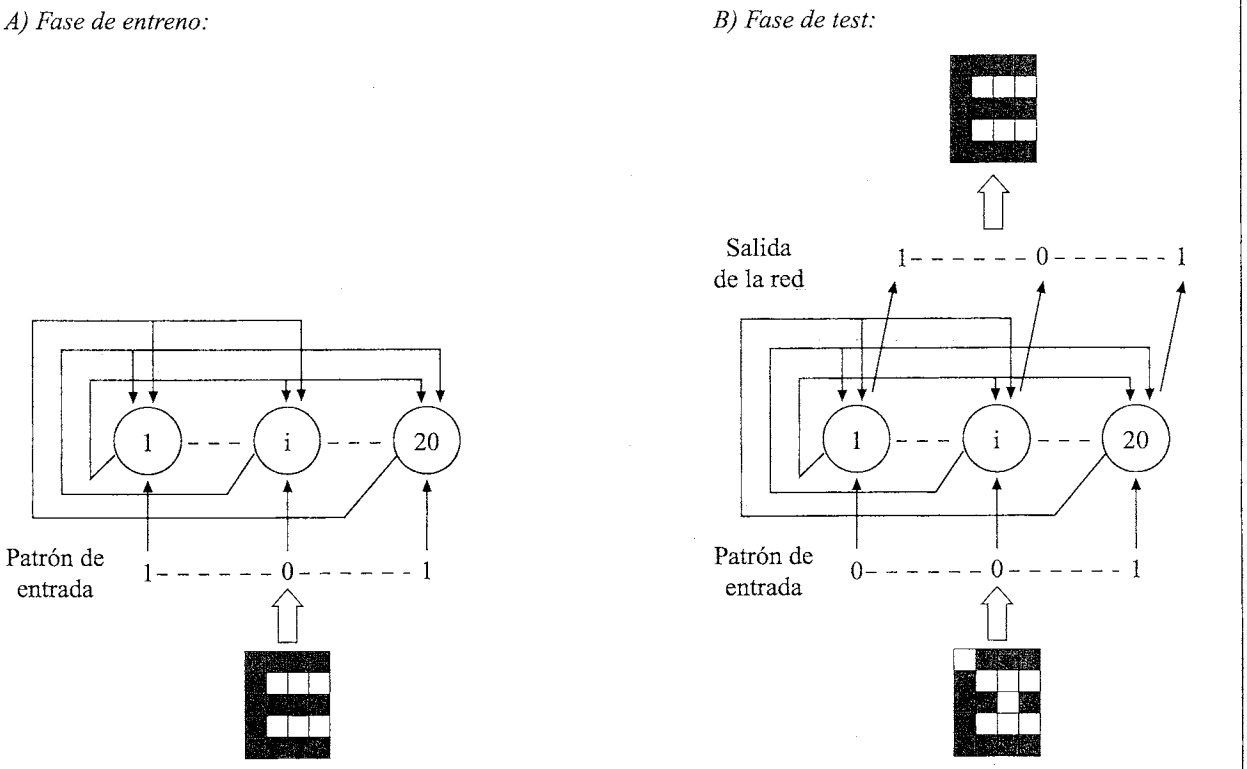
B) Fase de test:



sentación de ejemplares desconocidos, distorsionados o incompletos nos permitirá comprobar el grado de generalización que alcanza el modelo construido. En la figura 4B se muestra cómo la red proporciona una respuesta correcta ante un ejemplar incompleto de la vocal E que no había sido utilizado en la fase de entrenamiento.

Hemos visto un ejemplo de reconocimiento de patrones mediante la clasificación de cada figura en la categoría a la que pertenece. Se trata de un caso de red heteroasociativa. Ahora bien, podríamos estar interesados en entrenar la red para que aprendiera a asociar cada patrón o figura consigo misma. Como hemos visto, se trataría de un ejemplo de reconocimiento de patrones por autoasociación. Con fines ilustrativos, utilizaremos una red monocapa entrenada con aprendizaje no supervisado para realizar esta tarea, aunque en la práctica es más efectivo utilizar una red multicapa con aprendizaje supervisado. El número de neuronas de la red monocapa estará determinado por el número de píxeles que componen las figuras, en este caso es igual a 20; de forma que cada neurona se encargará de recibir y procesar un píxel. La fase de aprendizaje consistirá en el almacenamiento de los diferentes ejemplos de entrenamiento en los pesos de la red. Para ello, iremos presentando los ejemplos o patrones y la red irá modificando los pesos de forma iterativa hasta que alcancen una estabilidad. En la figura 5A se muestra este proceso para el caso de un ejemplar de la vocal E.

Figura 5. Entrenamiento y test de una red monocapa no supervisada para la reconstrucción de las vocales.



En la fase de test se demuestra la utilidad de este tipo de modelos. Permiten la reconstrucción de una determinada información de entrada que se presenta incompleta o distorsionada, proporcionando como salida la información almacenada más parecida. En la figura 5B se muestra cómo la red reconstruye en su salida la figura de una E a partir de su entrada incompleta.

Ventajas y limitaciones de las redes neuronales artificiales

Las RNA no son la panacea que permite resolver todos los problemas, sino que están orientadas a un determinado tipo de tareas. Podemos destacar cuatro características del problema o tarea que hacen aconsejable la utilización de las RNA (Martín del Brío y Sanz, 1997). Por una parte, no se dispone de un conjunto de reglas sistemáticas que describan completamente el problema. En cambio, sí disponemos de muchos ejemplos o casos (condición indispensable para poder aplicar las RNA). Por otra parte, los datos procedentes del problema son imprecisos, incoherentes o con ruido (como el ejemplo visto sobre el reconocimiento de la letra E). Por último, el problema es de elevada dimensionalidad, es decir, el número de variables de entrada es demasiado grande como para que un modelo convencional aprenda a solucionar el problema en un tiempo razonable.

Cuando no se dan estas circunstancias puede ser más aconsejable optar por solucionar el problema mediante un modelo derivado de la estadística o la Inteligencia Artificial. Por tanto, no debemos concebir las RNA como una alternativa, sino más bien como un complemento a los modelos convencionales ya establecidos.

Las RNA presentan una serie de ventajas frente a los modelos estadísticos. Una ventaja fundamental consiste en que los modelos neuronales normalmente no parten de restricciones respecto de los datos de partida (tipo de relación funcional entre variables), ni suele imponer presupuestos (como distribución gaussiana u otras). Por otra parte, como hemos comentado, la habilidad de las neuronas de calcular funciones de salida no lineales capacita a la red para resolver problemas complejos o no lineales. De este modo, en numerosas aplicaciones se están consiguiendo con RNA cotas de error mucho mejores que las proporcionadas por la estadística.

Respecto a las limitaciones que presentan las RNA, una de las más importantes consiste en que es difícil comprender la naturaleza de las representaciones internas generadas por la red para responder ante un

problema determinado. Es decir, no sabemos cómo el sistema interrelaciona las diferentes variables de entrada con los pesos de las conexiones entre neuronas para elaborar una solución (Rzempoluck, 1998). Esta limitación contrasta con los diferentes modelos estadísticos, los cuales permiten observar los parámetros o pesos relativos que el modelo otorga a cada una de las variables que intervienen en el modelo.

Con el fin de solventar esta limitación y así determinar qué es lo que la red ha aprendido, algunos autores (por ejemplo, Lisboa, Mehridehnavi y Martin, 1994) hacen uso de matrices de sensibilidad, las cuales permiten cuantificar la importancia que tiene cada variable de entrada sobre cada variable de salida de la red.

Realización de redes neuronales artificiales

La realización más simple e inmediata consiste en simular la red sobre un ordenador convencional mediante un software específico. Aunque de esta manera se pierde su capacidad de cálculo en paralelo, las prestaciones que ofrecen los ordenadores actuales resultan suficientes para resolver numerosos problemas prácticos, al permitir simular redes de tamaño considerable a una velocidad razonable. Esta constituye la manera más barata y directa de realizar una RNA. Por otra parte, no es necesario que cada investigador diseñe sus propios simuladores, pues existen numerosas aplicaciones comerciales que permiten la simulación de multitud de modelos neuronales (Hilera y Martínez, 1995; Martín del Brío y Sanz, 1997). Para consultar un listado actualizado de productos comerciales y de libre distribución, se recomienda visitar en internet el FAQ *(Frequent Asked Questions)* del grupo de noticias sobre RNA editado por Sarle (Sarle, 1998).

La alternativa a la simulación software en un ordenador, consiste en llevar a cabo la emulación hardware de la red neuronal, mediante el uso de procesadores especialmente diseñados para el trabajo con redes neuronales o mediante el diseño de circuitos específicos que reflejan con cierta fidelidad la arquitectura de la red (Hilera y Martínez, 1995; Martín del Brío y Sanz, 1997).

Aplicaciones generales

Las RNA son una tecnología computacional emergente que puede utilizarse en un gran número y variedad de aplicaciones. A continuación, proporcionamos un listado de aplicaciones de RNA en diferentes campos (McCord Nelson y Illingworth, 1991; Hilera y Martínez, 1995; Buscema, 1997):

–Biología

Estudio del cerebro
Obtención de modelos de retina

–Empresa

Identificación de candidatos para posiciones específicas
Reconocimiento de caracteres escritos
Predicción del rendimiento económico de las empresas

-Medio ambiente

Previsión del tiempo

–Finanzas

Previsión de la evolución de los precios
Valoración del riesgo de los créditos
Identificación de firmas

–Manufacturación

Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.)
Control de producción en líneas de proceso

–Medicina

Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalograma, análisis sanguíneo, cuestionarios, etc.)
Monitorización en cirugía
Predicción de reacciones adversas a los medicamentos
Lectores de rayos X

–Militares

Clasificación de las señales de radar
Creación de armas inteligentes
Reconocimiento y seguimiento de tiro al blanco
Detección de bombas

–Psicología y Psiquiatría

Modelización de procesos psicológicos básicos
Reconocimiento del habla (análisis e interpretación de frases habladas)
Diagnóstico de diversos trastornos (demencia, epilepsia, alcoholismo, etc.) en función de señales EEG
Clasificación de las fases del sueño
Diagnóstico psicológico
Predicción de rendimiento académico

REDES NEURONALES ARTIFICIALES APLICADAS A LA CONDUCTA ADICTIVA

En los apartados anteriores hemos visto que las RNA constituyen un modelo de procesamiento de la información robusto para la solución de problemas

complejos relacionados principalmente con el reconocimiento de patrones: clasificación, predicción y reconstrucción de ejemplares. Esta herramienta tecnológica ha sido aplicada muy recientemente en el campo de las adicciones. En este sentido, el Centro de Investigación Semeion de las Ciencias de la Comunicación (Roma, Italia), fundado y dirigido por Massimo Buscema, ha sido pionero en la aplicación de las RNA con el fin de prevenir y predecir la conducta adictiva. Los investigadores de dicho centro han construido diferentes modelos de red, los cuales pueden dividirse, siguiendo el esquema expuesto anteriormente, en dos grandes grupos: redes heteroasociativas y redes autoasociativas. Vamos a examinar cómo han aplicado estos dos tipos de redes al problema de las adicciones.

Buscema (1995) ha desarrollado un nuevo enfoque, denominado *Squashing Theory,* basado en el registro de un grupo de medidas biológicas, psicológicas y sociológicas con el fin de predecir, mediante un perceptrón multicapa supervisado, la conducta adictiva del sujeto. Más concretamente, se trata de entrenar una red para clasificar a los sujetos en dos posibles categorías, adicto (salida de la red = 1) o no adicto (salida de la red = 0), al presentarle a su entrada una serie de medidas obtenidas mediante cuestionario, susceptibles de ser predictoras del consumo de droga.

A continuación, se presentan las áreas específicas que deben ser evaluadas para la predicción de la conducta adictiva, de acuerdo con los principios de la *Squashing Theory* (Buscema, 1995):

a) Características académicas
b) Ocupación
c) Características y micro vulnerabilidad del padre, madre y hermanos
d) Condiciones de vida
e) Características sexuales y características de la pareja
f) Creencias religiosas
g) Estatus económico y gastos
h) Micro vulnerabilidad y estilo de vida relacionada con el alcohol y tabaco (no con adicción a drogas)
i) Problemas con la justicia
j) Amistades
k) Uso del tiempo libre
l) Características psicológicas
m) Micropercepciones de la familia y la pareja

Siguiendo este enfoque, Buscema (1995) seleccionó una muestra compuesta por tres grupos de sujetos. El primer grupo, 47 sujetos, se caracterizaba por estar recibiendo tratamiento por su adicción a la heroina. El segundo grupo, 94 sujetos, actuaba como grupo control y no había tenido ningún problema con las drogas. Estos dos grupos fueron etiquetados como casos prototípicos. Por último, el tercer grupo, 47 sujetos, estaba formado por sujetos que habían sido adictos a la heroina y habían dejado el tratamiento hacía al menos cinco años; por tal motivo, fueron etiquetados como casos inciertos. Para cada sujeto se registraron y codificaron numéricamente las variables de interés, determinándose su actual estatus de adicto o no a la heroina. La muestra total fue dividida aleatoriamente en casos de entrenamiento y casos de test. Obtenidos los pesos óptimos de la red neuronal a partir de los casos de entrenamiento, se comprobó la capacidad de predicción del modelo mediante la presentación de los casos de test. La red fue capaz de clasificar correctamente, en adicto o no adicto, el 92% de los casos prototípicos y el 80% de los casos inciertos.

Posteriormente, Buscema, Intraligi y Bricolo (1998) compararon el rendimiento de ocho modelos diferentes de red multicapa supervisada para la clasificación de los sujetos según su adicción o no a las drogas. Para ello, se usó una muestra compuesta por 223 sujetos adictos a la heroina y 322 sujetos control. La mitad de la muestra se utilizó para entrenar los diferentes modelos de red, la otra mitad sirvió para testar su rendimiento. La capacidad predictiva de los ocho modelos fue siempre superior al 91% en los casos de test, llegando a alcanzar, en algunos casos, el 97%.

Por su parte, Speri, Schilirò, Bezzetto et al. (1998) aplicaron los principios de la *Squashing Theory* al ámbito militar. Para ello, contaron con una muestra de 170 soldados compuesta por tres submuestras: 32 sujetos calificados de "normales", 24 sujetos altamente problemáticos y 114 sujetos con presunta o declarada adicción a las drogas. Se construyeron varias redes a partir de una configuración diferente de casos de entrenamiento y test. Todos los modelos mostraron unos resultados estables clasificando correctamente, en toxicómano o normal, al menos el 94 % de los casos de test. Posteriormente, se compararon las respuestas de las redes neuronales con las de una evaluación clínica estándar; el nivel de acuerdo fue superior al 70 % para los 170 casos.

Maurelli y Di Giulio (1998) compararon siete modelos diferentes de red neuronal para la predicción del grado de alcoholismo. La muestra estaba compuesta por 91 alcohólicos "moderados" y 22 alcohólicos "serios" que posteriormente fue dividida en casos de entrenamiento y test. El propósito de las redes consistía en dar como respuesta si el sujeto era alcohólico "moderado" o "serio" a partir de la entrada de cinco variables que representaban los resultados de varios tests biomédicos. Los resultados, a partir de los casos de test, fueron variados oscilando la capacidad de predicción de los modelos entre el 73 y el 86%. Posteriormente, se creó una nueva red, denominada

MetaNet, a partir de los cuatro modelos que habían obtenido mejores resultados. El modelo MetaNet alcanzó una capacidad de predicción del 93%.

Hasta ahora, hemos revisado los trabajos realizados por el equipo de Buscema sobre la utilización de redes neuronales heteroasociativas para la clasificación y/o predicción de la conducta adictiva.

Este equipo también ha utilizado modelos de red autoasociativa en el campo de las adicciones, creando recientemente una red autoasociativa, denominada red de satisfacción de restricciones (Rumelhart y McClelland, 1986), con el objeto de extraer los rasgos característicos relacionados con el consumo de droga. El aprendizaje de este tipo de red, compuesta por dos capas de igual tamaño –entrada y salida—, consiste en ir presentando en la capa de entrada los datos referidos a un grupo de sujetos –toxicómanos y no toxicómanos—, y en modificar los pesos de las conexiones de forma supervisada hasta que la capa de salida proporcione una información igual o similiar a la presentada a su entrada. Los datos que se presentan a la red harán referencia a las variables o características predictoras propuestas por la *Squashing Theory* y el estatus del sujeto como adicto o no. Ya vimos un proceso parecido en el almacenamiento de las vocales mediante una red monocapa no supervisada.

Una vez determinados los pesos de la red autoasociativa, podemos preguntar a la red qué rasgos prototípicos poseen los sujetos que pertenecen, por ejemplo, al grupo de toxicómanos (Buscema, Intraligi y Bricolo, 1998). Para ello, presentaremos como entrada el valor 0 (desactivado) para todas las neuronas que representan las diversas características del sujeto, excepto la neurona que representa el estatus de toxicomano; en este caso le presentamos el valor 1 (activado). La red proporcionará como salida los valores característicos de los sujetos toxicómanos para cada una de las variables predictoras.

Massini y Shabtay (1998) aplicaron este modelo de red en un centro de desintoxicación con metadona. A partir de una muestra compuesta por 69 pacientes del centro, la red neuronal permitió extraer las características prototípicas de los sujetos que había seguido con éxito el tratamiento de desintoxicación y los que no. Este procedimiento puede ser de gran utilidad ya que permite averiguar qué tratamiento será más adecuado en función del perfil del sujeto.

Para finalizar revisaremos los trabajos de un equipo de investigadores centrado en la predicción del alcoholismo a partir de respuestas psicofisiológicas. Así, Klöppel (1994) llevó a cabo un estudio preliminar con tres sujetos alcohólicos y tres sujetos control. A partir de la selección de dos sujetos de cada grupo, entrenó una red neuronal para clasificar los Potenciales Evocados (PE), previamente codificados numéricamente, de los sujetos en dos categorías: PE procedente de un sujeto alcohólico o PE procedente de un sujeto control. Los PE de los dos sujetos restantes actuaron como grupo de test. La red clasificó correctamente el 55.6% de los PE procedentes del sujeto alcohólico y el 89.4% de los PE procedentes del sujeto no alcohólico. Aunque los resultados no son muy buenos, el estudio muestra que la clasificación de los PE mediante una red neuronal es posible.

Recientemente, Winterer y sus colaboradores (Winterer, Klöppel, Heinz et al., 1998; Winterer, Ziller, Klöppel et al., 1998) se propusieron comprobar si a partir de los patrones electroencefalográficos cuantitativos (QEEG) se puede predecir, utilizando una red neuronal, la recaída de los sujetos alcohólicos al inicio del tratamiento. Se contó con una muestra de 78 pacientes alcohólicos que habían iniciado un tratamiento de desintoxicación. Se registraron los patrones QEEG de los sujetos siete días después de iniciado el tratamiento, determinándose tres meses más tarde dos posibles estatus: sujeto con recaída (49 sujetos) o sujeto abstinente (29 sujetos). Se entrenó un perceptrón multicapa supervisado para predecir si el sujeto había recaído o se había mantenido abstinente, ante la presentación del patrón QEEG del sujeto. La red fue capaz de predecir y/o clasificar correctamente el estatus del 85% de los casos de test. Con el objeto de comparar el rendimiento de la red con un modelo estadístico clásico, se aplicó el análisis discriminante lineal sobre las mismas variables. Este modelo clasificó correctamente el 75% de los casos de test. La aplicación del análisis discriminante no lineal (con polinomios de segundo orden) no mejoró este resultado. Aunque la red neuronal exhibió un rendimiento superior frente al análisis discriminante, estos resultados deben ser tomados con precaución debido al reducido número de sujetos con el cual se trabajó.

CONCLUSIONES

La primera parte de este trabajo ha pretendido ser una introducción general sobre el campo de las RNA evitando, de forma intencionada, la presentación de fórmulas matemáticas complejas –muy habituales incluso en documentos introductorios—, que no haría más que diezmar el número de lectores potenciales. Así, las RNA se presentan como una tecnología emergente de suma utilidad para la solución de problemas complejos en multitud de campos del conocimiento.

La segunda parte se ha centrado en la revisión de los trabajos que han aplicado los modelos de RNA en el no menos complejo campo de las adicciones. Los resultados obtenidos en los diferentes trabajos revisados confirman el papel de las RNA como una nueva y eficaz metodología para la descripción, prevención y predicción de la conducta adictiva. Así, hemos visto

que las redes heteroasociativas pueden predecir el estatus del sujeto como adicto o no adicto con un margen de error pequeño, en función de una serie de respuestas a un cuestionario. Por su parte, las redes autoasociativas permiten extraer los rasgos característicos de los sujetos adictos y no adictos, así como averiguar qué tratamiento será el más adecuado en función del perfil del sujeto. Aunque tales trabajos pueden ser calificados de preliminares, constituyen el punto de partida de futuras investigaciones que permitirán determinar el papel de las RNA en la predicción de la conducta adictiva. Así, se hace necesario realizar estudios comparativos respecto a los modelos estadísticos clásicos y utilizar muestras suficientemente grandes –no sólo muestras clínicas—, para poder extrapolar los resultados a la población. Estas investigaciones también nos permitirán averiguar si los buenos resultados obtenidos hasta el momento en las diferentes áreas de conocimiento se extienden al campo de las conductas adictivas.

REFERENCIAS BIBLIOGRÁFICAS

Alkon, D.L. (1989). Almacenamiento de memoria y sistemas neurales. *Investigación y Ciencia,* Septiembre, 14-23.

Arbib, M.A. (Ed.) (1995). *The handbook of brain theory and neural networks.* Cambridge, Mass.: MIT Press.

Arbib, M.A., Erdi, P. y Szentagothai, J. (1997). *Neural organization: structure, function and dynamics.* Cambridge, Mass.: MIT Press.

Bishop, C.M. (1995). *Neural networks for pattern recognition.* New York: Oxford University Press.

Bonilla, M. y Puertas, R. (1997). *Análisis de las redes neuronales: aplicación a problemas de predicción y clasificación financiera.* Valencia (España): Servei de Publicacions: Universitat de València.

Buscema, M. (1995). Squashing Theory: A prediction approach for drug behavior. *Drugs and Society, 8*(3-4), 103-110.

Buscema, M. (1997). A general presentation of artificial neural networks. I. *Substance Use & Misuse, 32*(1), 97-112.

Buscema, M. (1998). Artificial neural networks and complex systems. I. Theory. *Substance Use & Misuse, 33*(1), 1-220.

Buscema, M., Intraligi, M. y Bricolo, R. (1998). Artificial neural networks for drug vulnerability recognition and dynamic scenarios simulation. *Substance Use & Misuse, 33*(3), 587-623.

De Lillo, A. y Meraviglia, C. (1998). The role of social determinants on men's and women's mobility in Italy. A comparison of discriminant analysis and artificial neural networks. *Substance Use and Misuse, 33*(3), 751-764.

Duncan, J.C. (1997). A comparison of radial basis function and multilayer perceptron neural networks with linear multiple regression in cohort-survival based enrollment projection (Kent State University, 1996). *Dissertation Abstracts International, DAI-A 57/12,* 4995.

French, B.M., Dawson, M.R. y Dobbs, A.R. (1997). Classification and staging of dementia of the Alzheimer type: a comparison between neural networks and linear discriminant analysis. *Archives of Neurology, 54*(8), 1001-1009.

Hertz, J., Krogh, A. y Palmer, R.G. (1991). *Introduction to the theory of neural computation.* Redwood City, CA: Addison-Wesley.

Hilera, J.R. y Martínez, V.J. (1995). *Redes neuronales artificiales: Fundamentos, modelos y aplicaciones.* Madrid: Ra-Ma.

Jang, J. (1998). Comparative analysis of statistical methods and neural networks for predicting life insurers' insolvency (bankruptcy) (The University of Texas at Austin, 1997). *Dissertation Abstracts International, DAI-A 59/01,* 228.

Jefferson, M., Pendleton, N., Lucas, S. y Horan, M. (1997). Comparison of a genetic algorithm neural network with logistic regression for predicting outcome after surgery for patients with nonsmall cell lung carcinoma. *Cancer, 79*(7), 1338-1342.

Klöppel, B. (1994). Classification by neural networks of evoked potentials: A first case study. *Neuropsychobiology, 29*(1), 47-52.

Lisboa, P., Mehridehnavi, A. y Martin, P. (1994). The interpretation of supervised neural networks. pp. 11-17. En Lisboa, P. y Taylor, M. (Eds.). *Proceedings of the Workshop on Neural Network Applications and Tools.* Los Alamitos, CA: IEEE Computer Society Press.

Martín del Brío, B. y Sanz, A. (1997). *Redes neuronales y sistemas borrosos.* Madrid: Ra-Ma.

Massini, G. y Shabtay, L. (1998). Use of a constraint satisfaction network model for the evaluation of the methadone treatments of drug addicts. *Substance Use & Misuse, 33*(3), 625-656.

Masters, T. (1993). *Practical neural networks recipes in C++.* London: Academic Press.

Maurelli, G. y Di Giulio, M. (1998). Artificial neural networks for the identification of the differences between "light" and "heavy" alcoholics, starting from five nonlinear biological variables. *Substance Use & Misuse, 33*(3), 693-708.

McCord Nelson, M. y Illingworth, W.T. (1991). *A practical guide to neural nets.* Reading, MA: Addison-Wesley.

Pazos, A. (Ed.). (1996). *Redes de neuronas artificiales y algoritmos genéticos.* A Coruña: Universidade da Coruña, Servicio de Publicacions.

Ripley, B.D. (1996). *Pattern recognition and neural networks.* Cambridge: Cambridge University Press.

Rumelhart, D.E. y McClelland, J.L. (Eds.). (1986). *Parallel distributed processing: explorations in the microstructure of cognition.* Cambridge, Mass.: MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet.* New York: Springer-Verlag.

Sarle, W.S. (Ed.) (1998). *Neural network FAQ.* Periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: ftp://ftp.sas.com/pub/neural/FAQ.html.

Shekharan, R.A. (1997). Modeling pavement deterioration by regression and artificial neural networks (The University of Mississippi, 1996). *Dissertation Abstracts International, DAI-B 57/07,* 4578.

Shepherd, G.M. (1990). *The synaptic organization of the brain.* Oxford: Oxford Press.

Simpson, P.K. (Ed.) (1995). *Neural networks technology and applications: theory, technology and implementations.* New York: IEEE.

Smith, M. (1993). *Neural networks for statistical modeling.* New York: Van Nostrand Reinhold.

Speri, L., Schilirò, G., Bezzetto, A., Cifelli, G., De Battisti, L., Marchi, S., Modenese, M., Varalta, F. y Consigliere, F. (1998). The use of artificial neural networks methodology in the assessment of "vulnerability" to heroin use among army corps soldiers: A preliminary study of 170 cases inside the Military Hospital of Legal Medicine of Verona. *Substance Use & Misuse, 33*(3), 555-586.

Tommaso, M., Sciruicchio, V., Bellotti, R., Castellano, M., Tota, P., Guido, M., Sasanelli, G. y Puca, F. (1997). Discrimination between migraine patients and normal subjects based on steady state visual evoked potentials: discriminant analysis and artificial neural network classifiers. *Functional Neurology, 12*(6), 333-338.

Vohradsky, J. (1997). Adaptive classification of two-dimensional gel electrophoretic spot patterns by neural networks and cluster analysis. *Electrophoresis, 18*(15), 2749-2754.

Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psycometrika, 63*(1), 5-22.

West, P., Brockett, P. y Golden, L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science, 16*(4), 370-391.

Winterer, G., Klöppel, B., Heinz, A., Ziller, M., Dufeu, P., Schmidt, L.G. y Herrmann, W.M. (1998). Quantitative EEG (QEEG) predicts relapse in patients with chronic alcoholism and points to a frontally pronounced cerebral disturbance. *Psychiatry Research, 78*(1-2), 101-113.

Winterer, G., Ziller, M., Klöppel, B., Heinz, A., Schmidt, L.G. y Herrmann, W.M. (1998). Analysis of quantitative EEG with artificial neural networks and discriminant analysis: A methodological comparison. *Neuropsychobiology, 37*(1), 41-48.

2.2.
Predicción del consumo de éxtasis a partir de redes neuronales artificiales.

Predicción del consumo de éxtasis a partir de redes neuronales artificiales

PALMER POL, A.*; MONTAÑO MORENO, J.J.*; CALAFAT FAR, A.**

* Facultad de Psicología. Universidad de las Islas Baleares.
** IREFREA España

Enviar correspondencia a:
Alfonso Palmer Pol. Universidad de las Islas Baleares. Facultad de Psicología. Cra. de Valldemossa, km. 7,5. 07071 Palma de Mallorca (Baleares).
Teléfono 971173432; e-mail: alfonso.palmer@uib.es

Resumen:

El propósito del presente estudio fue mostrar cómo una red neuronal artificial (RNA) puede ser útil para predecir el consumo de éxtasis (MDMA). Más específicamente, se trata de desarrollar una red neuronal del tipo *backpropagation* capaz de discriminar entre quién consume éxtasis y quién no, a partir de las respuestas dadas por los sujetos a un cuestionario. La muestra estaba compuesta por 148 consumidores y 148 no consumidores de éxtasis. Se explican las diferentes fases llevadas a cabo para desarrollar la RNA: selección de las variables relevantes y preprocesamiento de los datos, división de la muestra en grupo de entreno, validación y test, entreno y evaluación del modelo de red, y análisis de sensibilidad. La eficacia de la RNA entrenada fue del 96.66%. El área bajo la curva ROC (*Receiver operating characteristic*) fue de 0.99440.0055 SE. Por otra parte, se pretende mostrar que las RNA no representan una "caja negra", sino que pueden dar información acerca del grado de influencia que tiene cada variable predictora sobre el consumo de éxtasis.

Palabras clave: *redes neuronales artificiales, éxtasis, factores de riesgo, clasificación de patrones.*

Abstract:

The purpose of this study was to show how an artificial neural network (ANN) can be useful to predict ecstasy (MDMA) consumption. More specifically, we tried to develop a backpropagation neural net capable to discriminate between who consumes ecstasy and who not, through the answers given by the subjects to a questionnaire. The sample was composed of 148 ecstasy consumers and 148 no consumers. We explain the diferent stages carried out to develop the ANN: selection of relevant variables and preprocessing of data, division of the sample into training, validation and test sets, training and evaluation of neural model, and sensitivity analysis. The accuracy of the ANN trained were 96.66%. The area under the ROC (Receiver operating characteristic) curve was 0.99440.0055 SE. On the other hand, we try to show that the ANN don't represent a "black box", but it can lead to useful insights into the roles played by different predictive variables in determining ecstasy consumption.

Key words: *artificial neural networks, ecstasy, risk factors, pattern classification.*

INTRODUCCIÓN

Las Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas (Hilera y Martínez, 1995). Consisten en un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. En los pesos se encuentra el conocimiento que tiene la RNA acerca de un determinado problema.

La utilización de las RNA puede orientarse en dos direcciones, bien como modelos para el estudio del sistema nervioso y los fenómenos cognitivos, bien como herramientas para la resolución de problemas prácticos como la clasificación de patrones y la aproximación de funciones. Desde esta segunda perspectiva, las RNA han sido aplicadas de forma satisfactoria en la predicción de diversos problemas en diferentes áreas de conocimiento —biología, medicina, economía, ingenie-

ría, psicología, etc.— (Arbib, 1995; Simpson, 1995; Arbib, Erdi y Szentagothai, 1997); obteniendo excelentes resultados respecto a los modelos derivados de la estadística clásica (West, Brockett y Golden, 1997; De Lillo y Meraviglia, 1998; Jang, 1998; Waller, Kaiser, Illian et al., 1998). La virtud de las RNA reside en su capacidad para aprender funciones complejas o no lineales entre variables sin necesidad de imponer presupuestos o restricciones de partida en los datos.

El uso de esta tecnología computacional es relativamente reciente en el problema de las conductas adictivas (Palmer y Montaño, 1999). En este sentido, el Centro de Investigación Semeion de las Ciencias de la Comunicación (Roma, Italia), fundado y dirigido por Massimo Buscema, puede ser considerado como pionero en la aplicación de las RNA en este campo. Los investigadores de dicho centro han construido diferentes modelos de red con el fin de predecir el consumo de droga —sobre todo heroína— (Buscema, 1995; Buscema, Intraligi y Bricolo, 1998; Maurelli y Di Giulio, 1998; Speri, Schilirò, Bezzetto et al., 1998), extraer las características prototípicas del sujeto adicto (Buscema, Intraligi y Bricolo, 1998) y así, determinar el tratamiento más adecuado en función de esas características (Massini y Shabtay, 1998). Aunque los resultados son preliminares, estos trabajos demuestran que los buenos resultados obtenidos hasta el momento en las diferentes áreas de conocimiento se pueden extender al campo de las adicciones.

Siguiendo la línea de investigación iniciada por el equipo de Buscema, nos hemos propuesto llevar a cabo la aplicación práctica de una red neuronal para la predicción del consumo de éxtasis (MDMA) y determinar la influencia de cada variable predictora sobre este tipo de conducta. Más concretamente, se trata de construir un modelo de red neuronal que a partir de las respuestas de los sujetos a un cuestionario, sea capaz de discriminar entre quién consume éxtasis y quién no.

En este sentido, el consumo de éxtasis y otros derivados de las feniletilaminas ha experimentado un aumento significativo en los últimos años aunque más recientemente dicho uso ha experimentado una cierta estabilización o incluso descenso desde los niveles de consumo tan altos que había (Plan Nacional sobre Drogas, 2000). En la encuesta escolar, tras haber crecido espectacularmente en el período 1994-96, se ha reducido en 1998 hasta situarse en los niveles que tenía en 1.994. También en la Encuesta domiciliaria sobre Drogas de 1.999 muestra que la proporción de españoles que habían consumido alguna vez éxtasis en el último año ha pasado a ser el 0,8% cuando en la anterior de 1.997 era del 1%. La importancia de este consumo ha provocado cierta alarma principalmente por la rapidez con que se ha producido su expansión y porque, aunque se trata de drogas cuyos efectos y toxicidad necesitan ser más investigados, existe suficiente evidencia acerca de su problemática (Calafat, Sureda y Palmer, 1997; Calafat, Stocco, Mendes et al, 1998).

Con el presente estudio, se averiguará si las RNA pueden ser empleadas en un futuro como herramientas de apoyo al profesional dedicado a la prevención del consumo de este tipo de sustancias.

MÉTODO

Sujetos

La muestra estaba formada por dos grupos de sujetos, 148 consumidores de éxtasis y 148 no consumidores de éxtasis. El muestreo fue intencional, encuestándose a los jóvenes en los lugares recreativos donde acudían, y se realizó en cinco países de la Comunidad Europea: España, Francia, Holanda, Italia y Portugal. A su vez la muestra se podía dividir en función del lugar donde se había pasado el cuestionario: un grupo de usuarios de discoteca y otro de estudiantes de Universidad. En la tabla 1 se presentan las características demográficas de los sujetos consumidores y no consumidores.

El grupo de consumidores se caracterizaba por ser consumidores habituales de éxtasis —consumían éxtasis más de una vez al mes. En general, los sujetos que formaban esta categoría eran además consu-

Tabla 1:
Características demográficas de los sujetos consumidores y no consumidores de éxtasis.

	Consumidores (n = 148)	**No consumidores** (n = 148)
Sexo		
Mujer	58	59
Varón	90	89
Edad	22.38* (4.15)	22.82* (4.30)
País		
España	48	34
Francia	21	29
Holanda	35	24
Italia	8	18
Portugal	36	43
Lugar		
Discoteca	108	69
Universidad	40	79

Nota: * Media y desviación estándar.

midores de otras sustancias como marihuana (n = 118), cocaína (n = 70), anfetaminas (n = 51), LSD (n = 44) y heroína (n = 7). Por su parte, el grupo de no consumidores que ha servido como grupo control se caracterizaba por no haber consumido nunca éxtasis ni ninguna otra sustancia ilegal.

Instrumentos

Con el objeto de determinar las características predictoras del consumo de éxtasis, se construyó un cuestionario compuesto por 25 ítems. Los ítems se podían agrupar en cinco categorías temáticas:

a) Demografía, relaciones con los padres y creencias religiosas
b) Ocio
c) Consumo
d) Opinión sobre el éxtasis
e) Personalidad

Las áreas exploradas por este cuestionario coinciden en gran medida con los principios de la *Squashing Theory*, enfoque desarrollado por Buscema (1995) y encaminado a la predicción de la conducta adictiva, mediante un modelo de red neuronal, a partir del registro de un conjunto de medidas biológicas, psicológicas y sociológicas.

La naturaleza de los ítems del cuestionario era variada. La mayoría eran variables cualitativas politómicas —p.e. "ocupación"—, pero había así mismo variables cualitativas dicotómicas —p.e. "¿eres creyente?"—, así como ítems de naturaleza ordinal —p.e. "estatus económico"—, e ítems de naturaleza cuantitativa —p.e. "puntuación en la escala de desviación social".

El modelo de red neuronal utilizado en la parte empírica de este trabajo fue simulado en un ordenador PC mediante el programa Neural Connection 2.0 (SPSS Inc., 1997a), el cual permite implementar el algoritmo de aprendizaje *backpropagation* en una arquitectura del tipo perceptrón multicapa.

Aplicación de la red neuronal

Resolver un problema mediante el uso de RNA supone aplicar una metodología que presenta aspectos comunes con las técnicas convencionales de modelado estadístico, pero también otros más particulares, que solamente se dan en el campo de las RNA. A continuación, se describen los pasos que se han seguido para la construcción de un modelo de red neuronal capaz de discriminar entre sujetos consumidores o no consumidores de éxtasis.

Selección de las variables relevantes y preprocesamiento de los datos

Para obtener una aproximación funcional óptima, se deben elegir cuidadosamente las variables a emplear. Más concretamente, de lo que se trata es de incluir en el modelo las variables predictoras que realmente predigan la variable dependiente, pero que a su vez no covaríen entre sí (Smith, 1993). La introducción de variables irrelevantes o que covaríen entre sí, puede provocar un sobreajuste innecesario en el modelo. Este fenómeno aparece cuando el número de parámetros o pesos de la red resulta excesivo en relación al problema a tratar y al número de patrones de entrenamiento disponibles. La consecuencia más directa del sobreajuste es una disminución sensible en la capacidad de generalización del modelo, es decir, la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento.

Teniendo en cuenta lo comentado, fue seleccionado un conjunto de 25 variables que permitían evaluar diferentes aspectos del sujeto, susceptibles de poder predecir el consumo de éxtasis. En la tabla 2 se proporciona una descripción de las variables predictoras utilizadas y la variable dependiente.

Una vez seleccionadas las variables que iban a formar parte del modelo, se procedió al preprocesamiento de los datos para adecuarlos a su tratamiento por la red neuronal. Para trabajar con el modelo de red neuronal aplicado en este estudio, el *backpropagation*, es muy aconsejable —aunque no imprescindible— conseguir que los datos posean una serie de cualidades (Masters, 1993; Martín del Brío y Sanz, 1997; SPSS Inc., 1997b; Sarle, 1998). Las variables deberían seguir una distribución normal o uniforme en tanto que el rango de posibles valores debería ser aproximadamente el mismo y acotado dentro del intervalo de trabajo de la función de activación empleada en las capas ocultas y de salida de la red neuronal.

Para adaptar nuestros datos a estas condiciones, se aplicó de forma satisfactoria una transformación logarítmica en las variables continuas que no seguían una distribución normal. A continuación, se acotó los valores de todas las variables predictoras al rango [-1, 1], límites de la función de activación que será utilizada por las neuronas de la capa oculta de la red. Este procedimiento permitió obtener mejores resultados que otros métodos de codificación comúnmente usados para el caso de variables cualitativas como, por ejemplo, los métodos 1-de-N y 1-de-N-1. Por su parte, la variable dependiente, estatus del sujeto, fue codificada como: -1 = no consumidor de éxtasis, 1 = consumidor de éxtasis.

Tabla 2: Descripción de las variables predictoras y la variable dependiente.

Variable	Alternativas de respuesta
Variables predictoras	
Demografía, padres y religión	
Estado civil	1: soltero/a 2: casado/a 3: vivo en pareja 4: otros
Nivel de estudios	1: primarios 2: bachiller 3: superiores
Ocupación	1: estudio 2: estudio y trabajo 3: trabajo eventual 4: trabajo fijo 5: servicio militar 6: parado 7: otros
Estatus económico	1: bajo 2: medio/bajo 3: medio 4: medio/alto 5: alto
¿Con quién vives?	1: padres/familia 2: conyuge/pareja 3: amigos 4: colegio/residencia 5: solo 6: otros
Relaciones con los padres	1: muy malas 2: bastante malas 3: regulares 4: bastante buenas 5: muy buenas
¿Eres creyente?	1: si 2: no
Ocio	
¿Vas a bares?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Vas a discotecas?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Vas a pubs?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Vas a cafés?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Vas a afters?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Vas a fiestas raves?	1: nunca 2: a veces 3: a menudo 4: casi siempre
¿Qué tipo de música prefieres?	1: house-bacalao 2: hardcore 3: hardcore-house 4: mellow-house 5: rock 6: pop 7: otros
Consumo	
¿Cuántos amigos toman éxtasis?	1: ninguno 2: pocos 3: la mitad 4: casi todos 5: todos
¿Has consumido alcohol este último mes?	1: si 2: no
¿Has consumido tabaco este último mes?	1: si 2: no
¿Te has emborrachado este último mes?	1: no 2: una vez al mes 3: varias veces al mes 4: alguna vez por semana 5: una vez por semana 6: cada día
Opinión sobre el éxtasis	
¿Crees que el éxtasis puede crear problemas?	1: no 2: sí, es ilegal 3: sí, después mal 4: sí, crea adicción 5: sí, amigos no toman 6: sí, efectos imprevisibles 7: sí, adulteración 8: sí, problemas con familia 9: otros
¿Cuál crees que es la razón para consumir éxtasis?	1: relajarse 2: disfrutar de bailar 3: bailar más tiempo 4: estar mejor con otros 5: olvidar los problemas 6: sentirse bien 7: mejor sexo 8: estimular los sentidos
Personalidad	
Escala de emoción y búsqueda de aventuras	Puntuación entre 0 y 10
Escala de búsqueda de experiencias	Puntuación entre 0 y 10
Escala de desinhibición	Puntuación entre 0 y 10
Escala de susceptibilidad al aburrimiento	Puntuación entre 0 y 10
Escala de desviación social	Puntuación entre 0 y 10
Variable dependiente	
Estatus de consumo de éxtasis	1: consumidor (más de una vez al mes) 2: no consumidor

Creación de los conjuntos de aprendizaje, validación y test

En la metodología de las RNA, a fin de encontrar la red que tiene la mejor ejecución con casos nuevos —es decir, que sea capaz de generalizar—, la muestra de datos es a menudo subdividida en tres grupos (Bishop, 1995; Ripley, 1996): entrenamiento, validación y test.

Durante la etapa de aprendizaje de la red, los pesos son modificados de forma iterativa de acuerdo con los valores del grupo de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red y la salida deseada por el usuario. Sin embargo, como ya se ha comentado, cuando el número de parámetros o pesos es excesivo en relación al problema —fenómeno del sobreajuste—, el modelo se ajusta demasiado a las particularidades irrelevantes presentes en los patrones de entrenamiento en vez de ajustarse a la función subyacente que relaciona entradas y salidas, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos.

Para evitar el problema del sobreajuste, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje. Durante el aprendizaje la red va modificando los pesos en función de los datos de entrenamiento y de forma alternada se va obteniendo el error que comete la red ante los datos de validación. De este modo, podemos averiguar cuál es el número de pesos óptimo, en función de la arquitectura que ha tenido la mejor ejecución con los datos de validación. Como se verá más adelante, mediante el grupo de validación también se puede determinar el valor de otros parámetros que intervienen en el aprendizaje de la red.

Por último, si se desea medir de una forma completamente objetiva la eficacia final del sistema construido, no deberíamos basarnos en el error que se comete ante los datos de validación, ya que de alguna forma, estos datos han participado en el proceso de entrenamiento. Se debería contar con un tercer grupo de datos independientes, el grupo de test el cuál proporcionará una estimación insesgada del error de generalización.

En el presente estudio, se obtuvieron estos tres conjuntos de datos mediante una asignación aleatoria de los 296 sujetos que formaban la muestra. Así, se contó con 176 sujetos de entrenamiento –de los cuales 88 eran consumidores y 88 eran no consumidores de éxtasis—, 60 sujetos de validación –de los cuales 30 eran consumidores y 30 eran no consumidores de éxtasis—, y 60 sujetos de test –de los cuales 30 eran consumidores y 30 eran no consumidores de éxtasis.

Entrenamiento de la red neuronal

El modelo de red neuronal empleado ha sido una arquitectura del tipo perceptrón multicapa entrenada mediante la regla de aprendizaje *backpropagation* (propagación del error hacia atrás) (Rumelhart, Hinton y Williams, 1986). El perceptrón multicapa está formado por una capa de entrada, una capa de salida y una o más capas ocultas o intermedias; la información se transmite desde la capa de entrada hasta la capa de salida y cada neurona está conectada con todas las neuronas de la siguiente capa. La utilización del algoritmo *backpropagation* o alguna de sus múltiples variantes supone alrededor del 80% de las aplicaciones que se realizan con RNA (Caudill y Butler, 1992).

El funcionamiento de una red de este tipo consiste en el aprendizaje de un conjunto de pares de entradas y salidas de información dados como ejemplo, empleando un ciclo de propagación-adaptación compuesto por dos fases. En nuestro caso, la red debe aprender a relacionar los valores de las variables predictoras con el correspondiente estatus de consumo del sujeto. En la fase de propagación, se presenta a la capa de entrada de la red los valores de las 25 variables predictoras correspondientes a un sujeto de entrenamiento, esta información se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado obtenido con la salida que se desea obtener —-1 si el sujeto es no consumidor y 1 si el sujeto es consumidor —, y se calcula el error que comete la neurona de la capa de salida. En la fase de adaptación, este error se propaga hacia atrás (de ahí el nombre que recibe), capa por capa, recibiendo cada neurona un error que describe su aportación relativa al error global que comete la red. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presenten los valores del mismo sujeto, la salida esté más cerca de la deseada, es decir, el error disminuya.

A continuación, se expone la expresión matemática de la regla de modificación de pesos descrita (para una explicación más detallada, consultar: Rumelhart, Hinton y Williams, 1986):

$$\Delta w_{ji}(n+1) = \varepsilon\, \delta_{pj}\, x_{pi} + \eta \Delta w_{ji}(n)$$

donde

w_{ji} = peso entre la neurona i y la neurona j

n = número de iteración

ε = tasa de aprendizaje (junto al momento controla el tamaño del cambio de los pesos en cada iteración)

δ_{pj} = error de la neurona j para el patrón p

x_{pi} = salida de la neurona i para el patrón p

η = momento

Una vez que se han presentado todos los patrones de entrenamiento, se procede a actualizar el valor de los pesos de la red, completándose así un ciclo de aprendizaje o iteración. Con este proceso, se pretende minimizar la siguiente función de error:

$$E = \frac{1}{2} \sum_p \sum_k (d_{pk} - x_{pk})^2$$

donde

d_{pk} = salida deseada de la neurona de salida k para el patrón p

x_{pk} = salida real de la neurona de salida k para el patrón p

Es decir, el error que comete la red neuronal se obtiene calculando simplemente la diferencia entre la salida deseada por el usuario y la salida proporcionada por la red para cada patrón o sujeto de entrenamiento.

Antes de comenzar este proceso de aprendizaje, se debe asignar unos valores iniciales a los pesos de umbral y de conexión entre neuronas. Se adoptó el procedimiento común de asignar estos valores de

forma aleatoria dentro del rango [-0.5, 0.5] con una distribución uniforme (SPSS Inc., 1997a). Por otra parte, existe una serie de parámetros cuyo valor no se puede conoce *a priori* dado un problema, sino que deben ser determinados mediante ensayo y error. La utilización de un grupo de validación ayudará a conocer el valor óptimo de cada uno de estos parámetros: arquitectura de la red, valor de la tasa de aprendizaje y del momento, y función de activación de las neuronas de la capa oculta y de salida. Así, la configuración de parámetros que obtenga el menor error ante los datos de validación, será la seleccionada para pasar a la fase de test.

Respecto a la arquitectura de la red, se sabe que para la mayoría de problemas prácticos bastará con utilizar una capa de entrada, una oculta y una de salida (Funahashi, 1989; Hornik, Stinchcombe y White, 1989). El número de neuronas de la capa de entrada está determinado por el número de variables predictoras. Cada neurona de entrada tiene como misión recibir y transmitir a la siguiente capa, el valor de una de estas variables. Por su parte, el número de neuronas de la capa de salida está determinado, en tareas de clasificación, por el número de categorías o clases que tiene el problema. En nuestro caso, la única neurona de salida dará como resultado el valor -1 si el sujeto es no consumidor y 1 si el sujeto es consumidor. Por último, no existe una receta que indique el número óptimo de neuronas en la capa oculta para un problema dado. Recordando el problema del sobreajuste, se debe usar el mínimo número de neuronas ocultas con las cuales la red rinda de forma adecuada (Masters, 1993; Smith, 1993; Rzempoluck, 1998). Así, evaluando el rendimiento de diferentes arquitecturas en función de los resultados obtenidos con el grupo de validación, se seleccionó una capa oculta compuesta por dos neuronas.

Los valores de la tasa de aprendizaje *(ε)* y el momento *(η)* tienen un papel crucial en el proceso de entrenamiento de una red neuronal, ya que controlan el tamaño del cambio de los pesos en cada iteración. Se deben evitar dos extremos: un ritmo de aprendizaje demasiado pequeño puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar con una configuración de pesos poco eficiente; en cambio, un ritmo de aprendizaje demasiado grande puede conducir a inestabilidades en la función de error o a saturar las neuronas de la red. Por tanto, se recomienda elegir un ritmo de aprendizaje lo más grande posible sin que provoque grandes oscilaciones. En general, el valor de la tasa de aprendizaje suele estar comprendida entre 0.05 y 0.5, mientras que el valor del momento suele ser aproximadamente igual a 0.9 (Rumelhart, Hinton y Williams, 1986). En nuestro estudio, los mejores resultados se obtuvieron con unos valores de $\varepsilon = 0.3$ y $\eta = 0.8$. Esta configuración de valores permitió alcanzar la convergencia —es decir, hasta que el valor de los pesos permanece estable—, en 1200 iteraciones o ciclos de aprendizaje, momento en que se decidió parar el entrenamiento.

Por último, la función de activación es la función que se aplica a la entrada neta de la neurona para obtener un valor de salida. La entrada neta es la suma del producto de cada señal que recibe de las neuronas de la capa anterior por el valor del peso que conecta ambas neuronas, menos el umbral de la neurona (el umbral es considerado como un peso que conecta con una neurona ficticia con valor de salida igual a 1):

$$net_j = \sum_{i=1}^{N} w_{ji} x_i - \theta_j$$

El algoritmo *backpropagation* exige que la función de activación sea continua y, por tanto, derivable para poder obtener el error o valor delta de las neuronas ocultas y de salida. Se disponen de dos formas básicas que cumplen esta condición: la función lineal (o identidad) y la función sigmoidal. Sin embargo, es absolutamente imprescindible, para aprovechar la capacidad de las RNA de aprender relaciones complejas o no lineales entre variables, la utilización de funciones no lineales al menos en las neuronas de la capa oculta (Rzempoluck, 1998). En este sentido, los mejores resultados se obtuvieron utilizando la función sigmoidal tangente hiperbólica (con límites entre -1 y 1) para las neuronas de la capa oculta y la función lineal para la neurona de la capa de salida.

En consonancia con nuestros resultados, los estudios experimentales realizados muestran que la utilización de valores bipolares (positivos y negativos) en las funciones de activación acelera considerablemente el entrenamiento de la red frente a la utilización de valores binarios como es el caso de la función sigmoidal logística (con límites entre 0 y 1) (Fahlman, 1988; Kalman y Kwasny, 1992; Fausett, 1994).

La figura 1 muestra el modelo de red neuronal utilizado en este estudio, la arquitectura estaba compuesta por 25 neuronas en la capa de entrada, dos neuronas en la capa oculta y una neurona en la capa de salida.

Evaluación del rendimiento de la red neuronal

La evaluación del rendimiento de una RNA entrenada se realiza mediante el uso de un grupo de datos que no haya participado en el proceso de aprendizaje, el grupo de test. Con esto, se persigue obtener algún tipo de medida que permita estimar la capacidad de generalización del modelo. En este sentido, existe un amplio abanico de medidas de rendimiento (Masters, 1993): media cuadrática del error, funciones de coste, matrices de confusión, índices de sensibilidad y especificidad, etc..

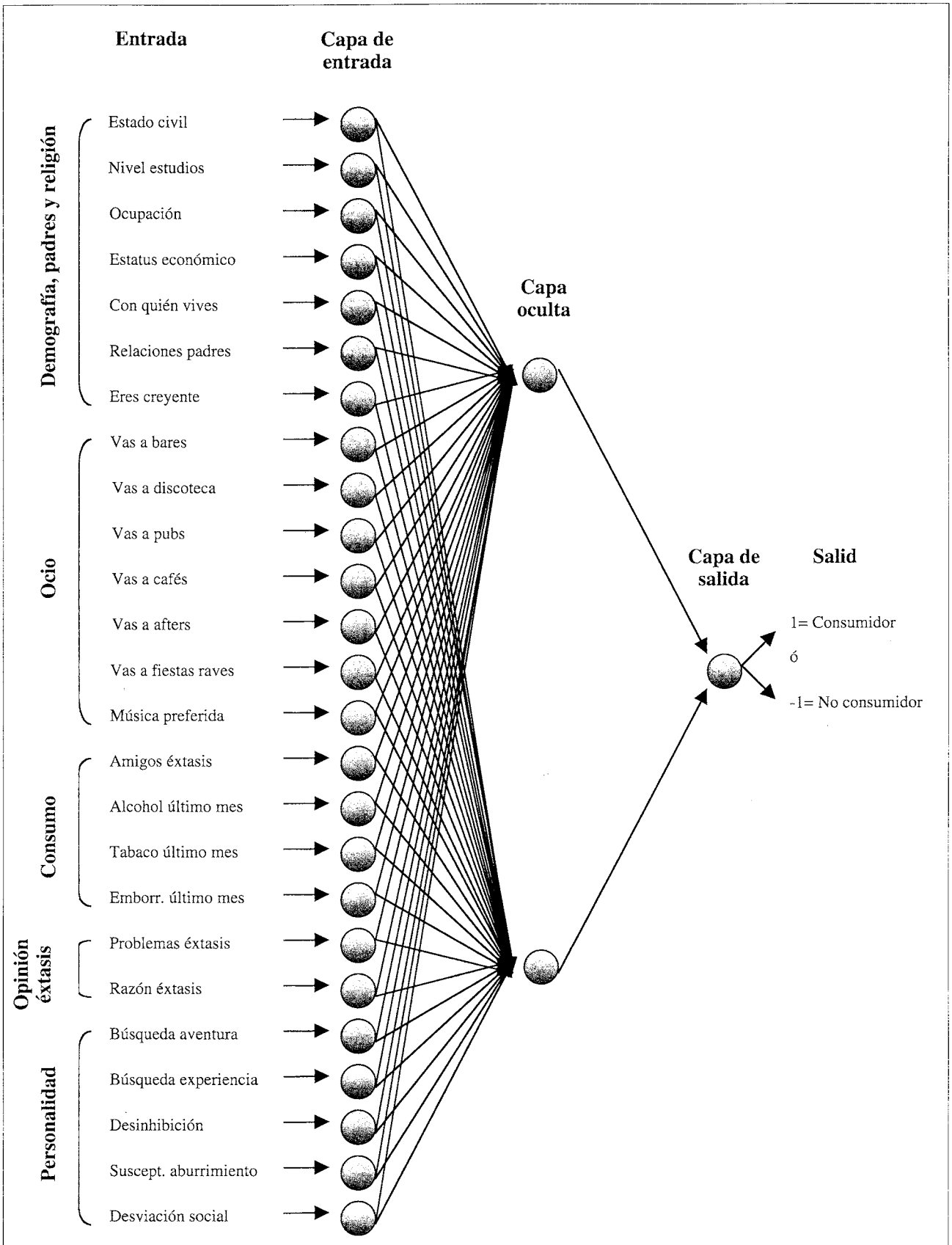


Figura 1. Modelo de red neuronal utilizado en el estudio.

En nuestro estudio, la evaluación del rendimiento se realizó a partir de los índices de sensibilidad, especificidad y eficacia, y del análisis de curvas ROC (*Receiver operating characteristic*).

Se recuerda al lector que la sensibilidad de un instrumento diagnóstico es, en nuestro caso, el porcentaje de consumidores que son clasificados correctamente —verdaderos positivos. Por su parte, la especificidad es el porcentaje de no consumidores que son clasificados correctamente –verdaderos negativos. Por último, a raíz de los dos índices anteriores, la eficacia es el porcentaje de sujetos (consumidores y no consumidores) correctamente clasificados.

El análisis de curvas ROC se originó a principios de los años 50 en el seno de la teoría de detección electrónica de señales (TDS), y se ha destacado en los últimos años como una medida precisa y válida para evaluar la precisión diagnóstica de un instrumento (Swets, 1973, 1988). Las curvas ROC poseen dos ventajas fundamentales respecto a los tradicionales índices de sensibilidad, especificidad y eficacia: son independientes del punto de corte elegido y de la prevalencia —en nuestro caso, de la proporción de sujetos consumidores. Para nuestros fines, la curva ROC consistiría en la representación gráfica del porcentaje de verdaderos positivos (sensibilidad) en el eje de ordenadas, contra el porcentaje de falsos positivos (1-especificidad) en el eje de abcisas, para diferentes puntos de corte aplicados sobre la salida que proporciona la red neuronal —un valor cuantitativo aproximadamente entre -1 y 1. Los verdaderos positivos serían sujetos consumidores clasificados por la red como consumidores, mientras que los falsos positivos serían sujetos no consumidores clasificados por la red como consumidores. En este tipo de análisis, la medida de resumen más utilizada es el área total bajo la curva ROC. Esta medida se interpreta como la probabilidad de clasificar correctamente un par de sujetos —uno consumidor y otro no consumidor—, seleccionados al azar, fluctuando su valor entre 0.5 y 1. El área bajo la curva ROC de un instrumento inútil es 0.5, reflejando que al ser utilizado clasificamos correctamente un 50% de individuos, idéntico porcentaje al obtenido utilizando simplemente el azar. Por el contrario, el área bajo la curva ROC de un instrumento perfecto es 1, ya que permite clasificar sin error el 100% de sujetos.

Análisis de sensibilidad

Una de las críticas más importantes que se han lanzado contra el uso de RNA trata sobre lo difícil que es comprender la naturaleza de las representaciones internas generadas por la red para responder ante un problema determinado (De Laurentiis y Ravdin, 1994; Rzempoluck, 1998). A diferencia de los modelos estadísticos clásicos, no es tan evidente conocer en una red la importancia que tiene cada variable predictora sobre la/s variable/s dependiente/s. Sin embargo, esta percepción acerca de las RNA como una compleja "caja negra", no es del todo cierta. De hecho, han surgido diferentes intentos por interpretar los pesos o parámetros del modelo (Masters, 1993), de los que el más ampliamente utilizado es el denominado análisis de sensibilidad (Hashem, 1992; Lisboa, Mehridehnavi y Martin, 1994). Se debe advertir al lector que el término sensibilidad utilizado en el apartado anterior no tiene ningún tipo de relación con el término análisis de sensibilidad utilizado en esta ocasión. Recordemos que la sensibilidad es el porcentaje de verdaderos positivos de un instrumento diagnóstico, mientras que el análisis de sensibilidad es un procedimiento para conocer el efecto o influencia de cada variable predictora sobre la/s variable/s dependiente/s.

El método más común para realizar un análisis de sensibilidad consiste en fijar el valor de todas las variables de entrada a su valor medio e ir variando el valor de una de ellas a lo largo de todo su rango, con el objeto de observar el efecto que tiene sobre la salida de la red. Siguiendo este método, se fue registrando los cambios que se producían en la salida de la red cada vez que se aplicaba un pequeño incremento *n* —incrementos de un 2%—, en una variable de entrada. Se propuso como objetivo cuantificar la influencia que tiene cada variable de entrada. Pensamos que la simple suma de los cambios producidos proporcionaría una medida intuitiva de sensibilidad. Esta medida representaría el efecto relativo que tiene una variable de entrada sobre la salida de la red. Así, un valor cercano a 0 indicaría poco efecto o sensibilidad; a medida que se fuese alejando de 0, indicaría que el efecto va aumentando. Esta medida de sensibilidad se obtuvo mediante la siguiente expresión:

$$S_{ik} = \sum_{n}^{N} |x_{kn} - x_{kmin}|$$

donde

- S_{ik} = medida de sensibilidad de la variable de entrada *i* sobre la salida *k*
- x_{kn} = valor de la salida *k* obtenido con el incremento *n* en la variable de entrada *i*
- x_{kmin} = valor de la salida *k* obtenido con el valor mínimo posible de la variable de entrada *i*

RESULTADOS

Rendimiento del modelo de red neuronal

El modelo de red neuronal finalmente seleccionado obtuvo unos resultados excelentes a partir del

grupo de test –recordemos que este grupo estaba compuesto por 30 sujetos consumidores y 30 sujetos no consumidores de éxtasis. Así, estableciendo un punto de corte igual a cero en la salida de la red —las salidas negativas eran consideradas como "no consumidores" y las positivas como "consumidores"—, únicamente dos sujetos, uno de cada grupo, fueron incorrectamente clasificados. Por tanto, los valores —en términos de porcentaje—, de la sensibilidad, especificidad y eficacia de la red fueron todos del 96.66%. Por su parte, el área total bajo la curva ROC (gráfico 1) dio como resultado 0.9944±0.0055 SE, aportando más datos a favor de la eficacia predictora del modelo entrenado.

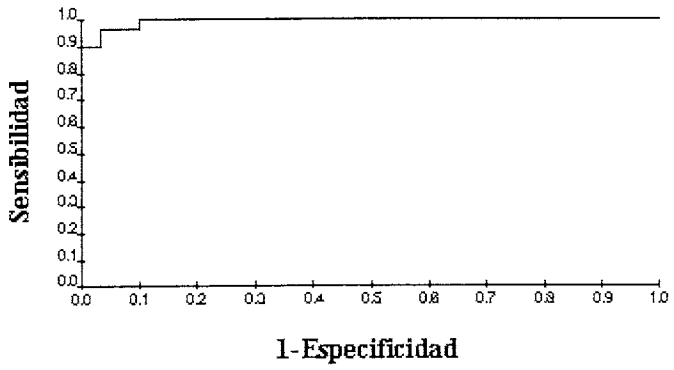


Gráfico 1. Curva ROC del modelo de red a partir del grupo de test.

Rendimiento de los submodelos de red neuronal

Una vez demostrado el excelente rendimiento del modelo de red entrenado, se quiso examinar la capacidad predictora de cada una de las cinco categorías temáticas —demografía, padres y religión, ocio, consumo, opinión sobre el éxtasis y personalidad—, sobre el consumo de éxtasis. Para ello, se crearon cinco submodelos de red, cada uno entrenado a partir de las variables que formaban una categoría temática. Las condiciones de entrenamiento y evaluación fueron las mismas que las usadas para el modelo general de red utilizado inicialmente.

En la tabla 3 se presentan los índices de rendimiento de los cinco submodelos de red a partir del grupo de test.

Tabla 3: Indices de rendimiento de los cinco submodelos de red a partir del grupo de test.

Categoría	Sensibilidad	Especificidad	Eficacia	Area ROC*
Demografía, padres y religión	80.00	66.66	73.33	0.80 (0.05)
Ocio	90.00	93.33	91.66	0.96 (0.02)
Consumo	90.00	80.00	85.00	0.95 (0.02)
Opinión sobre el éxtasis	46.66	93.33	70.00	0.74 (0.06)
Personalidad	90.00	70.00	80.00	0.88 (0.04)

Nota: * Area bajo la curva ROC y error estándar.

Los valores de sensibilidad, especificidad, eficacia y área bajo la curva ROC indican que las dos categorías con mayor poder predictivo son las de ocio (91.66% de eficacia y 0.96 de área ROC) y consumo (85% de eficacia y 0.95 de área ROC). La categoría de personalidad alcanza un valor predictivo muy satisfactorio con una eficacia del 80% y un área ROC de 0.88. Por último, las categorías de demografía, padres y religión (73.33% de eficacia y 0.80 de área ROC), y opinión sobre el éxtasis (70% de eficacia y 0.74 de área ROC) son las que presentan menor poder predictivo. Aunque la primera de ellas presenta una sensibilidad del 80% y la segunda presenta una especificidad del 93.33%.

En el gráfico 2 se muestra la curva ROC de cada uno de los cinco submodelos de red a partir del grupo de test.

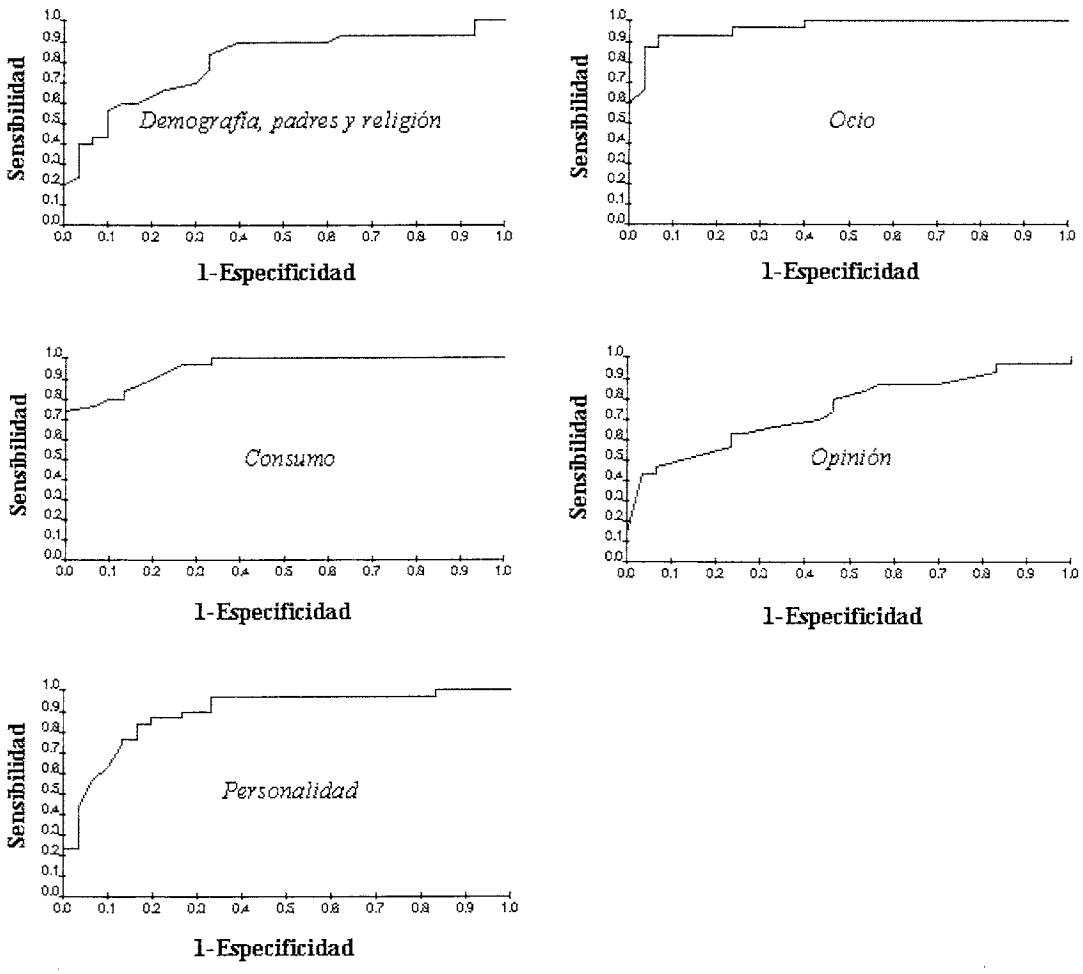


Gráfico 2: Curvas ROC de los cinco submodelos de red a partir del grupo de test.

Análisis de sensibilidad

A partir del modelo general inicialmente entrenado, se obtuvo el valor de la medida de sensibilidad para cada variable predictora sobre el consumo de éxtasis. En la tabla 4 se presentan estos valores ordenados de mayor a menor. Así, los primeros valores de la tabla corresponden a las variables de entrada con más influencia o relación con la salida de la red –estatus de consumo del sujeto.

Así, se puede observar que las variables que tienen mayor influencia en el consumo de éxtasis son: la cantidad de amigos/as que consumen éxtasis (S = 58.93), el consumo de tabaco (S = 43.18), la frecuencia en asistir a afters (S = 41.24), el estatus económico (S = 34.22), el tipo de música preferida (S = 26.50) y la frecuencia en asistir a fiestas raves (S = 26.21). Estos resultados concuerdan con los obtenidos al evaluar el rendimiento de los diferentes submodelos, es decir, las variables de ocio y consumo son las que tienen mayor efecto sobre el consumo de éxtasis.

Tabla 4: Medida de sensibilidad de las variables predictoras sobre el consumo de éxtasis.

Variable predictora	Sensibilidad
¿Cuántos amigos toman éxtasis?	58.93
¿Has consumido tabaco este último mes?	43.18
¿Vas a afters?	41.24
Estatus económico	34.22
¿Qué tipo de música prefieres?	26.50
¿Vas a fiestas raves?	26.21
Escala de desinhibición	22.69
Relaciones con los padres	22.47
¿Cuál crees que es la razón para consumir éxtasis?	20.21
¿Eres creyente?	16.94
Escala de desviación social	15.90
¿Crees que el éxtasis puede crear problemas?	15.89
¿Has consumido alcohol este último mes?	12.31
Escala de susceptibilidad al aburrimiento	11.03
¿Vas a pubs?	10.19
Estado civil	9.42
Escala de emoción y búsqueda de aventuras	9.27
¿Vas a cafés?	7.34
Escala de búsqueda de experiencia	6.84
Ocupación	4.93
¿Con quién vives?	4.14
¿Vas a bares?	3.89
¿Tes has emborrachado este último mes?	2.80
Nivel de estudios	2.77
¿Vas a discotecas?	1.08

Por último, para obtener una información más completa, la medida de sensibilidad debería ir acompañada de la correspondiente representación gráfica. En el gráfico 3 se muestra la representación gráfica del análisis de sensibilidad de las 12 primeras variables de la tabla 3, es decir, las 12 variables que muestran más influencia o relación con la salida de la red.

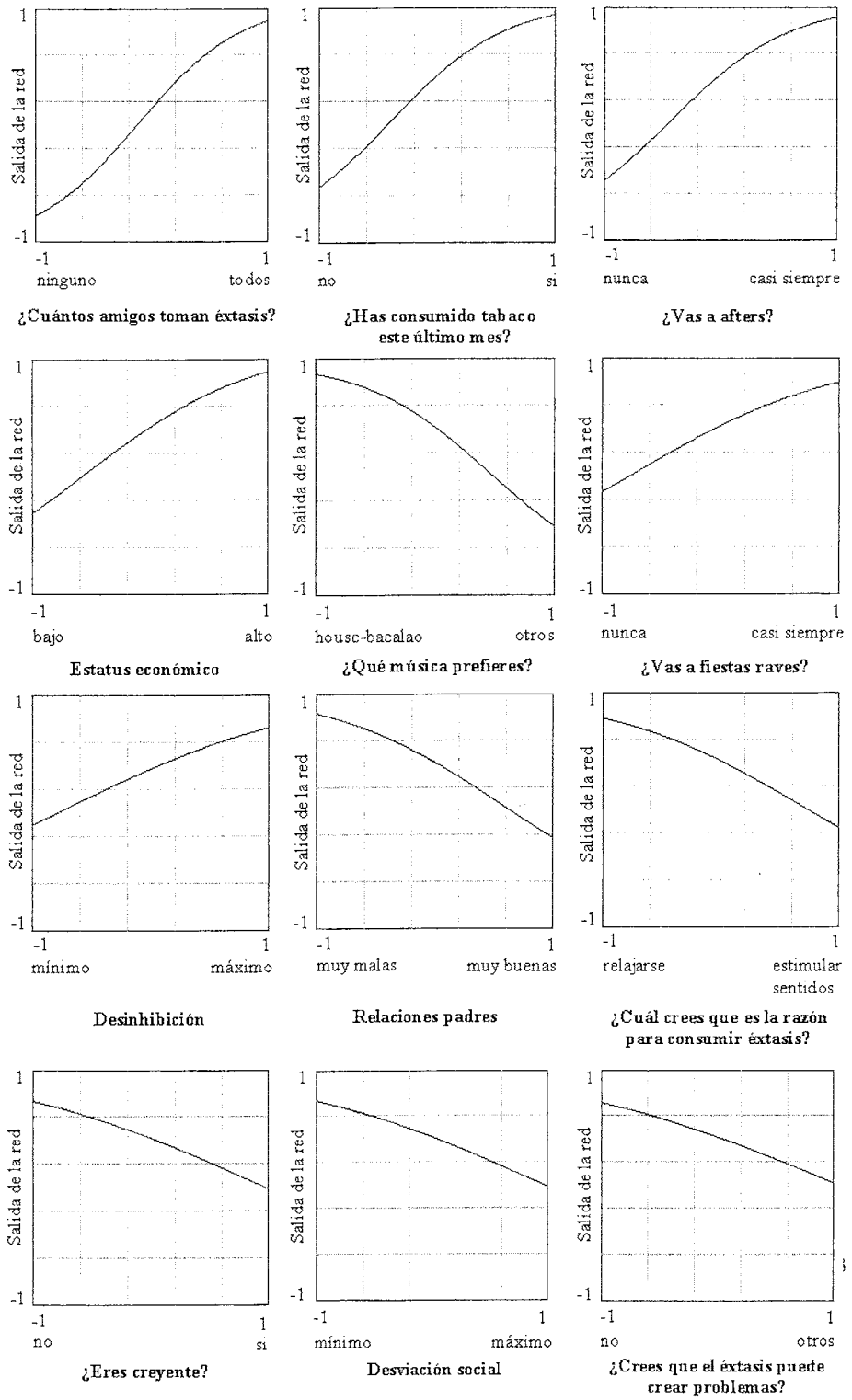


Gráfico 3: Representación gráfica del análisis de sensibilidad de las 12 variables predictoras con mayor relación o influencia sobre el consumo de éxtasis.

A modo de ejemplo, se puede observar en el citado gráfico el efecto o relación que mantiene la variable "*¿Cuántos amigos toman éxtasis?*" y la salida de la red –estatus de consumo del sujeto. Cuando dicha variable toma el valor –1 (ningún amigo toma éxtasis), la salida de la red es aproximadamente igual a –1 (no consumidor de éxtasis). A medida que se incrementa el valor de la variable de entrada (-0.5 = pocos, 0 = la mitad, 0.5 = casi todos consumen éxtasis), el valor de la salida de la red también va incrementándose. Finalmente, cuando la variable toma el valor 1 (todos mis amigos consumen éxtasis), la salida de la red es aproximadamente igual a 1 (consumidor de éxtasis). Por tanto, se puede decir que el número de amigos que consumen éxtasis está relacionado con la salida de la red y, por tanto, es un predictor del estatus de consumo del sujeto.

CONCLUSIONES

Se ha presentado una RNA capaz de predecir el consumo de éxtasis a partir de las respuestas dadas a un cuestionario, con un grado de eficacia del 96.66%. Esto significa que conociendo las respuestas del sujeto a esas 25 preguntas, se puede anticipar si ese sujeto es consumidor o no de éxtasis, con un margen de error muy pequeño. Los resultados obtenidos, en nuestro estudio, son acordes con los obtenidos por el equipo de Buscema. Así, por ejemplo, Buscema, Intraligi y Bricolo (1998) desarrollaron varios modelos de red neuronal para la predicción de la adicción a la heroína. La eficacia de los diferentes modelos fue siempre superior al 91%, llegando a alcanzar, en algunos casos, el 97%. Por su parte, Maurelli y Di Giulio (1998) obtuvieron un modelo de red capaz de predecir el grado de alcoholismo de un sujeto, a partir de los resultados de varios tests biomédicos, con una capacidad de predicción del 93%. Todos estos resultados muestran que las excelentes cualidades exhibidas por las RNA en las diferentes disciplinas, se extienden al campo de las conductas adictivas.

Por otra parte, se ha pretendido mostrar, en contra de la concepción tradicional, que los pesos de un modelo de red neuronal pueden dar información acerca del grado de influencia de las variables de entrada sobre la salida de la red. De este modo, se ha mostrado que cuanto más alto sea el índice de sensibilidad () de una determinada variable de entrada, más relación o influencia ejercerá sobre la salida de la red –estatus de consumo o no consumo del sujeto. Intentos como el nuestro se encuentran en los trabajos de Modai, Saban, Stoler et al. (1995), los cuales identificaron mediante un análisis de sensibilidad los factores de buen pronóstico ante la aplicación de un tratamiento en pacientes psiquiátricos. Por su parte, Kashani, Nair, Rao et al. (1996), con un esquema similar identificaron los factores asociados a las autoexpectativas negativas en adolescentes.

Por último, pensamos que los desarrollos futuros deberían ir encaminados hacia la aplicación de RNA en el resto de conductas relacionadas con el uso y abuso de sustancias —anfetaminas, cocaína, marihuana, etc.—, con el objeto de identificar los factores que influyen en cada una de estas conductas mediante el uso de índices de sensibilidad robustos. Los resultados de estos desarrollos podrían facilitar información importante a la hora de confeccionar programas de prevención de la conducta adictiva.

REFERENCIAS BIBLIOGRÁFICAS

Arbib, M.A. (Ed.) (1995). *The handbook of brain theory and neural networks*. Cambridge, Mass.: MIT Press.

Arbib, M.A., Erdi, P. y Szentagothai, J. (1997). *Neural organization: structure, function and dynamics*. Cambridge, Mass.: MIT Press.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Buscema, M. (1995). Squashing Theory: A prediction approach for drug behavior. *Drugs and Society, 8*(3-4), 103-110.

Buscema, M., Intraligi, M. y Bricolo, R. (1998). Artificial neural networks for drug vulnerability recognition and dynamic scenarios simulation. *Substance Use & Misuse, 33*(3), 587-623.

Calafat, A., Sureda, M.P. y Palmer, A. (1997). Características del consumo de éxtasis en una muestra de universitarios y usuarios de discoteca. *Adicciones, 9*(4), 529-555.

Calafat, A.; Stocco, P.; Mendes, et al (1998) *Characteristics and Social Representation of Ecstasy in Europe*. Palma de Mallorca. IREFREA.

Caudill, M. y Butler, C. (1992). *Understanding neural networks: Computer explorations*. Cambridge, MA: MIT Press.

De Laurentiis, M. y Ravdin, P.M. (1994). A technique for using neural network analysis to perform survival analysis of censored data. *Cancer Letters, 77*, 127-138.

De Lillo, A. y Meraviglia, C. (1998). The role of social determinants on men's and women's mobility in Italy. A comparison of discriminant analysis and artificial neural networks. *Substance Use and Misuse, 33*(3), 751-764.

Fahlman, S. (1988). *An empirical study of learning speed in back-propagation networks*. Tech. Rep. CMU-CS-88-162.

Fausett, L. (1994). *Fundamentals of neural networks*. New Jersey: Prentice-Hall.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks, 2*, 183-192.

Hashem, S. (1992). Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. *International Joint Conference on Neural Networks*, 419-424.

Hilera, J.R. y Martínez, V.J. (1995). *Redes neuronales artificiales: Fundamentos, modelos y aplicaciones*. Madrid: Ra-Ma.

Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

Jang, J. (1998). Comparative analysis of statistical methods and neural networks for predicting life insurers´ insolvency (bankruptcy) (The University of Texas at Austin, 1997). *Dissertation Abstracts International, DAI-A, 59/01*, 228.

Kalman, B.L. y Kwasny, S.C. (1992). Why tanh? Choosing a sigmoidal function. *International Joint Conference on Neural Networks*, 578-581.

Kashani, J.H., Nair, S.S., Rao, V.G., Nair, J. y Reid, J.C. (1996). Relationship of personality, environmental, and DICA variables to adolescent hopelessness: a neural network sensitivity approach. *Journal American Children and Adolescent Psychiatry, 35*(5), 640-645.

Lisboa, P., Mehridehnavi, A. y Martin, P. (1994). The interpretation of supervised neural networks. *Proceedings of the Workshop on Neural Network Applications and Tools*, 11-17.

Martín del Brío, B. y Sanz, A. (1997). *Redes neuronales y sistemas borrosos*. Madrid: Ra-Ma.

Massini, G. y Shabtay, L. (1998). Use of a constraint satisfaction network model for the evaluation of the methadone treatments of drug addicts. *Substance Use & Misuse, 33*(3), 625-656.

Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press.

Maurelli, G. y Di Giulio, M. (1998). Artificial neural networks for the identification of the differences between "light" and "heavy" alcoholics, starting from five nonlinear biological variables. *Substance Use & Misuse, 33*(3), 693-708.

Modai, I., Saban, N.i., Stoler, M., Valevski, A. y Saban, N. (1995). Sensitivity profile of 41 psychiatric parameters determined by neural network in relation to 8-week outcome. *Computers in Human Behavior, 11*(2), 181-190.

Palmer, A. y Montaño, J.J. (1999). ¿Qué son las redes neuronales artificiales?. Aplicaciones realizadas en el ámbito de las adicciones. *Adicciones, 11*(3), 243-255.

Plan Nacional sobre Drogas (2.000). *Informe nº 3. Observatorio español sobre drogas*. Madrid: Plan Nacional sobre Drogas.

Ripley, B.D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet*. New York: Springer-Verlag.

Sarle, W.S. (Ed.) (1998). *Neural network FAQ*. Periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: ftp://ftp.sas.com/pub/neural/FAQ.html.

Simpson, P.K. (Ed.) (1995). *Neural networks technology and applications: theory, technology and implementations*. New York: IEEE.

Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.

Speri, L., Schilirò, G., Bezzetto, A., Cifelli, G., De Battisti, L., Marchi, S., Modenese, M., Varalta, F. y Consigliere, F. (1998). The use of artificial neural networks methodology in the assessment of "vulnerability" to heroin use among army corps soldiers: A preliminary study of 170 cases inside the Military Hospital of Legal Medicine of Verona. *Substance Use & Misuse, 33*(3), 555-586.

SPSS Inc. (1997a). *Neural Connection 2.0* [Programa para ordenador]. SPSS Inc. (Productor). Chicago: SPSS Inc. (Distribuidor).

SPSS Inc. (1997b). *Neural Connection 2.0: User's Guide* [Manual de programa para ordenadores]. Chicago: SPSS Inc.

Swets, J.A. (1973). The relative operating characteristic in psychology. *Science, 182*, 990-1000.

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*, 1285-1293.

Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psycometrika, 63*(1), 5-22.

West, P., Brockett, P. y Golden, L. (1997). A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science, 16*(4), 370-391.

2.3.
Las redes neuronales artificiales en Psicología: un estudio bibliométrico.

Metodología de las Ciencias del Comportamiento 3(1), 53-64

Las redes neuronales artificiales en psicología: un estudio bibliométrico

B. Cajal**,[1] R. Jiménez**, J.M. Losilla*, J.J. Montaño**, J.B. Navarro*, A. Palmer**, A. Pitarque***, M. Portell*, M.F. Rodrigo***, J.C. Ruíz*** y J. Vives*

GRUPO ERNAP

* *Universitat Autònoma de Barcelona,*
** *Universitat de les Illes Balears,*
*** *Universitat de València*

Resumen

En el presente estudio se describe la creación de una base de datos bibliográfica sobre Redes Neuronales Artificiales (RNA) en el ámbito de la Psicología y los principales resultados que se desprenden del análisis bibliométrico realizado a partir de la misma. El análisis bibliométrico incidió en los siguientes aspectos: evolución temporal de la productividad, autores, revistas y editoriales más productivas, análisis de materias, aplicaciones más frecuentes, papel que desempeñan las RNA en las diferentes áreas de la Psicología y análisis de los estudios centrados en la comparación entre modelos estadísticos y RNA. La base de datos creada y el análisis bibliométrico realizado ha servido como fondo documental para el Grupo ERNAP (Grupo para el Estudio de las Redes Neuronales Artificiales en Psicología), el cual ha iniciado una serie de líneas de investigación en RNA con unos resultados muy prometedores.
PALABRAS CLAVE: *Redes Neuronales Artificiales, Psicología, Análisis Bibliométrico*

Abstract

ARTIFICIAL NEURAL NETWORKS IN PSYCHOLOGY: A BIBLIOMETRIC STUDY. In this study we describe the creation of a bibliographic database about Artificial Neural Networks (ANN) in the field of Psychology and the main results of the bibliometric analysis applied to the database. The bibliometric analysis was focused on: temporary evolution of productivity, authors, reviews and most productive editorials, contents analysis, most frequent applications, role of the ANN in the different fields of Psychology and analysis of the studies that compare statistical models and ANN. The database created and the bibliometric analysis carried out were useful as background documentation for the ERNAP (Estudio de las Redes Neuronales Artificiales en Psicología), who has initiated a set of researches with promising results.
KEY WORDS: *Artificial Neural Networks, Psychology, Bibliometric Analysis*

Tras la constitución del Grupo para el Estudio de las Redes Neuronales Artificiales en Psicología (Grupo ERNAP), compuesto por miembros del Área de Metodología de distintas universidades (Universidad Autónoma de Barcelona, Universidad de las Islas Baleares y Universidad de Valencia), sus miembros nos propusimos como primer objetivo la creación de una base de datos que recopilase el mayor número posible de trabajos sobre Redes Neuronales Artificiales (RNA) en el ámbito de la Psicología y de la Metodología de las Ciencias del Comportamiento. También nos interesaba analizar trabajos pertenecientes a otros ámbitos (como medicina, biología, ingeniería, etc.), ya que podrían aportarnos nuevas ideas para posteriormente ser aplicadas en el campo de la Psicología y la Metodología.

La generación de tal base de datos permitiría no sólo contar con un fondo documental para su consulta, sino también la posibilidad de extraer información sistematizada - autores, revistas y editoriales más productivas, análisis de materias,

[1]Dirección postal del primer autor: Berta Cajal Blasco. Universitat de les Illes Balears. Facultad de Psicología. Ctra. de Valldemossa, Km. 7.5. 07071 Palma de Mallorca (España). *E-mail:* dpsbcb0@clust.uib.es

áreas de aplicación, etc. -, mediante la aplicación de un análisis bibliométrico sobre la base de datos. El estudio realizado consta de dos fases principales: (1) selección y recopilación de información, y (2) análisis bibliométrico.

Selección y recopilación de información

El primer paso que llevamos a cabo fue la confección de un listado de palabras clave o descriptores con el fin de emprender una búsqueda sobre el campo de las RNA en diferentes bases de datos. Con el objetivo de que dicha búsqueda fuera lo más exhaustiva posible seleccionamos las siguientes palabras clave y descriptores generales: *neural networks, connectionism, parallel processing* y *parallel distributed processing.*

Mediante este conjunto de palabras clave se inició una búsqueda en ocho bases de datos. La selección de estas bases de datos se llevó a cabo en dos fases. En una primera fase, se realizó un listado de las bases de datos disponibles en las universidades implicadas en el estudio, o bien, de acceso libre en Internet. En una segunda fase, se seleccionaron las bases de datos disponibles que se adaptaban mejor al objetivo del estudio, esto es, la creación de una base de datos sobre RNA en Psicología y en otros campos de aplicación.

A partir de las bases de datos seleccionadas, se recopilaron un total de 11.003 registros. Estos registros fueron importados al gestor bibliográfico EBLA 3.0 (Losilla, 1997), el cual permite realizar de forma rápida y fácil todas las tareas relacionadas con la manipulación de referencias bibliográficas (catalogación y búsqueda flexible de la información, importación y exportación de referencias, generación de listados, fichas y estadísticas, etc.). Posteriormente realizamos un proceso de filtrado con el fin de descartar aquellos registros que, a pesar de haber sido inicialmente incluidos, no versaban sobre RNA. También se procedió a la eliminación de los registros repetidos o duplicados, esto es, los registros presentes en más de una base de datos. De esta forma, la base de datos resultante constó de un total de 7.891 registros o referencias. La Tabla 1 muestra el listado de las bases de datos consultadas, su descripción y los resultados generales obtenidos. Cabe mencionar que todas las bases de datos consultadas fueron revisadas hasta Junio de 1998. El año de inicio de la búsqueda en cada base de datos se determinó en función de la disponibilidad de la base de datos y, por supuesto, en función de la existencia de registros sobre RNA.

A partir de la base de datos resultante se obtuvo el listado de los 10 descriptores o palabras clave que aparecían con más frecuencia. Como cabía esperar el descriptor más frecuente es Neural Networks con 3.095 apariciones. Un dato menos obvio es que dentro de este listado se encontraban dos descriptores propios del campo de la Psicología: Cognitive Processes, con 278 apariciones, y Memory, con 232 apariciones. Este primer dato ya revelaba el papel de la Psicología en el estudio de las RNA.

Análisis bibliométrico

Sobre la base de datos generada se aplicaron un conjunto de procedimientos derivados de la investigación bibliométrica y basados en análisis estadísticos descriptivos y sociométricos, que permitieron descubrir información valiosa acerca de la produc-

ción científica en el campo de las RNA como, por ejemplo, autores más productivos, revistas y editoriales dominantes, líneas de investigación actuales o utilización de RNA en las diferentes áreas de la Psicología y Metodología, etc. El lector interesado en la metodología del análisis bibliométrico puede consultar los excelentes trabajos de Carpintero (1980), Méndez (1986), Sancho (1990), Alcain (1991), Romera (1992), Ferreiro (1993) y Amat (1994). A continuación se detallan los principales resultados obtenidos.

Tabla 1. Bases de datos consultadas.

Base de datos	Descripción	Editor	Años revisados	Nº registros
Dissertation Abstracts	Recoge citas (con resumen) de aproximadamente un millón de tesis doctorales y "masters" desde 1861 de unas 500 universidades.	University Microfilms International	1980-1998	1251
Eric	Comprende citas (con resumen) de educación del Educational Resources Information Center del US Department of Education. Recoge las fuentes: RIE y CIJE. Contiene información desde 1966.	Dialog Information Services	1980-1998	137
ISBN	Base de datos sobre libros registrados en España desde 1972.	Agencia Española del ISBN	1972-1998	18
Library of Congress	Catálogo de la Librería del Congreso norteamericana. Mantiene un catálogo, accesible desde internet, con las publicaciones incluidas en su registro informatizado desde 1968 (con más de 4,5 millones de registros).	Library of Congress	1968-1998	1277
Medline	Base de datos de la US National Library of Medicine. Incluye las citas (con resumen) de los artículos publicados en más de 3.000 revistas biomédicas, un 75% de las cuales están en lengua inglesa.	Cambridge Scientific Abstracts	1980-1998	2810
PsycLit	Base de datos de la American Psychological Association. Equivale a la publicación Psychological Abstracts. Indexa más de 1.300 revistas especializadas en psicología y ciencias del comportamiento. Recoge materiales relativos a psicología, psiquiatría, sociología, antropología, educación, etc. Contiene información desde 1974.	SilverPlatter Information, Inc.	1981-1998	2201
Sociofile	Base de datos que equivale a la publicación Sociological Abstracts. Incluye referencias (con resumen) sobre sociología aparecidas desde 1974. Incorpora la base de datos SOPODA que contiene información desde 1980.	SilverPlatter Information, Inc.	1974-1998	64
Teseo	Contiene información (cita y resumen) sobre tesis doctorales leídas en universidades españolas desde 1976.	Ministerio de Educación, Cultura y Deporte	1976-1998	133
			Total	7891

Evolución temporal de la productividad

Con el análisis de la evolución temporal de la productividad podemos averiguar si el interés por un tema ha crecido, ha declinado o se mantiene estable durante un período de tiempo. En este sentido, la Tabla 2 muestra la evolución temporal de la productividad que se da en nuestra base de datos desde 1949 hasta 1998 - debemos recordar que las bases de datos consultadas fueron revisadas hasta Junio de 1998 -.

Tabla 2. Evolución temporal de la productividad.

Años	Tipo de publicación				Nº total	%	% acum.
	Libros	Artículos	Tesis	Informes			
1949-80	64	0	1	0	65	0.82	0.82
1981	9	2	1	0	12	0.15	0.97
1982	12	3	0	0	15	0.19	1.16
1983	3	10	2	1	16	0.20	1.36
1984	15	3	1	2	21	0.26	1.62
1985	25	26	4	3	58	0.73	2.36
1986	26	27	5	5	63	0.79	3.15
1987	46	40	13	3	102	1.29	4.44
1988	61	85	30	4	180	2.28	6.72
1989	85	169	40	1	295	3.73	10.45
1990	117	280	74	3	474	6.00	16.45
1991	139	285	137	4	565	7.16	23.61
1992	154	386	145	4	689	8.73	32.34
1993	152	515	180	4	851	10.78	43.12
1994	185	607	195	8	995	12.60	55.72
1995	174	611	202	4	991	12.55	68.27
1996	158	708	178	4	1048	13.28	81.55
1997	139	688	140	1	968	12.26	93.81
1998	34	413	36	0	483	6.12	100
Total	1598	4858	1384	51	7891	100.0	0

Como se puede observar, el grado de producción o interés por las RNA es mínimo hasta aproximadamente la mitad de los años 80. A partir de esa fecha el interés comienza a aumentar, primero de forma tímida, y a partir de 1990 de forma significativa alcanzando un pico de producción situado en el año 1996, que cuenta con 1.048 publicaciones. En el año 1994 se da la mayor producción de libros e informes técnicos, mientras que en los años 1995 y 1996 se da la mayor producción de tesis doctorales y artículos, respectivamente. La Figura 1 refleja la evolución temporal descrita.

Sin duda, el creciente interés por las RNA, que se manifiesta de forma palpable a principios de los 90, está relacionado con un hecho fundamental en la historia de las RNA, a saber: la publicación de Parallel Distributed Processing (Procesamiento Distribuido en Paralelo o PDP) (Rumelhart, McClelland y el grupo PDP, 1986), obra que se ha llegado a conocer como la "biblia" del nuevo paradigma conexionista. De esta publicación se pueden destacar dos importantes aportaciones. En primer lugar, se propone un marco general que identifica de forma sistematizada las características comunes de la mayor parte de modelos PDP y redes neuronales (Rumelhart, Hinton y McClelland, 1986). En segundo lugar, se presenta un nuevo algoritmo, denominado backpropagation, que permite el aprendizaje en redes feedforward con unidades ocultas (Rumelhart, Hinton y Williams, 1986), superando las limitaciones que presentaba la regla de aprendizaje asociada al Perceptrón simple de Rosenblatt (1958). En la actualidad, la utilización del algoritmo backpropagation o alguna de

sus múltiples variantes supone alrededor del 80% de las aplicaciones que se realizan con RNA (Caudill y Butler, 1992). Para explicar el interés despertado por las RNA no podemos olvidar tampoco, entre otras, las aportaciones de Anderson, Silverstein, Ritz y Jones (1977), Carpenter y Grossberg (1986), Fukushima (1988), Hopfield (1982, 1984) y Kohonen (1984).

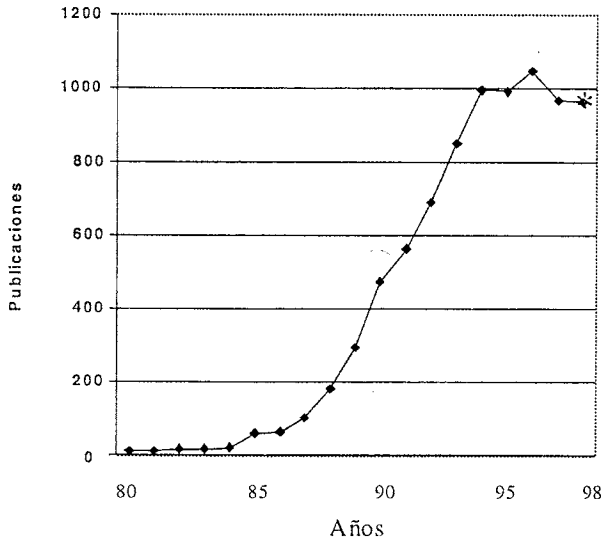


Figura 1: Evolución temporal de la productividad[2]

Autores, revistas y editoriales más productivas

En la Tabla 3 se muestra el listado de los diez autores más productivos. Más concretamente, se proporciona para cada autor, el área principal de investigación, el número de trabajos en los que el autor firma solo, el número de trabajos en los que el autor firma en colaboración con otros autores y, por último, el número total de trabajos firmados. Como podemos observar, en primer lugar se halla Stephen Grossberg, con 69 trabajos en nuestra base de datos, quien en la propia literatura de RNA es considerado como uno de los autores más prolíficos. Otros autores destacados son David Rumelhart, James McClelland, Geoffrey Hinton y Terrence Sejnowski, miembros fundadores del ya comentado grupo PDP (Parallel Distributed Processing) que popularizaron el uso de las RNA a finales de los años 80, llevando a cabo importantes aportaciones en el campo de la algoritmia y el modelado de procesos psicológicos.

Examinando con más detalle las líneas de investigación desarrolladas por los autores más prolíficos descubrimos que, de forma casi unánime, tales autores se centran en el modelado de procesos psicológicos y fisiológicos. Así, las líneas de investigación más destacadas versan sobre el modelado de procesos de memoria (McClelland y Reggia), reconocimiento de palabras (McClelland), función hipocampal (Schmajuk), percepción visual (Grossberg), cortex visual (Grossberg, Cohen y Sejnowski) y reflejo vestibulo-ocular (Sejnowski).

[2]El asterisco que corresponde al año 1998 indica un valor estimado, ya que se ha registrado hasta junio de 1998.

Tabla 3: Autores más productivos.

Autor	Tema principal	Trabajos del autor	Trabajos de 2 autores	Trabajos de 3 o más autores	Total
Stephen Grossberg	Modelado Procesos Psicofisiológicos	8	35	26	69
Terrence J. Sejnowski	Modelado Procesos Fisiológicos	1	23	18	42
James L. McClelland	Modelado Procesos Psicológicos	5	15	15	35
Nestor A. Schmajuk	Modelado Procesos Psicofisiológicos	3	15	11	29
Jonathan D. Cohen	Modelado Procesos Psicofisiológicos	1	8	18	27
James A. Reggia	Modelado Procesos Psicofisiológicos	1	11	14	26
John G. Taylor	Manuales	9	11	4	24
Geoffrey E. Hinton	Algoritmos	5	9	9	23
Michael A. Arbib	Modelado Procesos Fisiológicos	6	9	7	22
Donald E. Rumelhart	Manuales	2	11	6	19

Por otra parte, en la Tabla 4 se muestra el listado de las diez revistas más productivas. Para cada revista se proporciona la temática, el número de artículos y el porcentaje que representa respecto al total de artículos registrados en la base de datos. Este listado viene encabezado por *Neural Networks* y por *Neural Computation*, dos revistas centradas en algoritmos y modelos de aprendizaje para redes neuronales. Las revistas dedicadas al modelado de procesos psicológicos y fisiológicos - *Biological Cybernetics, Connection Science* y *Cognition* - también tienen un papel destacado en la base de datos. Por otra parte, teniendo en cuenta que el número de revistas registradas en nuestra base de datos es de 968, cabe señalar que las diez revistas más productivas representan el 27% de la producción total de artículos, lo que significa que la producción de artículos está concentrada en un número reducido de revistas.

Por último, en la Tabla 5 se muestra el listado de las diez editoriales más productivas. Para cada editorial se proporciona el número de libros y el porcentaje que representa respecto al total de libros registrados en la base de datos. Este listado viene encabezado por Springer, MIT Press, Lawrence Erlbaum y Kluwer Academic. Las editoriales se centran en la producción de libros introductorios a las RNA y de manuales de consulta. Al igual que en el caso de las revistas, la producción de libros está concentrada en un número reducido de editoriales. Así, teniendo en cuenta que el número de editoriales registradas en nuestra base de datos es de 315, las diez editoriales más productivas representan el 53% de la producción total de libros.

Tabla 4: Revistas más productivas.

Revista	Tema principal	Número de artículos	Porcentaje
Neural Networks	Algoritmos	335	6.89
Neural Computation	Algoritmos	332	6.83
Biological Cybernetics	Modelado de procesos fisiológicos	188	3.86
International Journal of Neural Systems	Algoritmos	172	3.54
Connection Science	Modelado de procesos psicológicos	61	1.25
IEEE Transactions on Systems, Man and Cybernetics	Algoritmos	56	1.15
Neurocomputing: An International Journal	Algoritmos	49	1.00
IEEE Transactions on Biomedical Engineering	Aplicaciones en Medicina	44	0.90
Cognition	Modelado de procesos psicológicos	42	0.86
Artificial Intelligence in Medicine	Aplicaciones en Medicina	41	0.84
Total		1320	27.12

Tabla 5: Editoriales más productivas.

Editorial	Número de libros	Porcentaje
Springer	219	13.70
MIT Press	151	9.44
Lawrence Erlbaum	101	6.32
Kluwer Academic	95	5.94
World Scientific	61	3.81
North Holland	51	3.19
Academic Press	46	2.87
Wiley	45	2.81
IEE Computer Society Press	43	2.69
Elsevier Science	40	2.50
Total	852	53.27

Análisis de materias

Todos los registros de la base de datos fueron clasificados en una de entre ocho materias o áreas temáticas. Esta labor clasificatoria no sólo permitiría realizar labores de filtrado y discriminación de los registros en función de su temática, sino también analizar el grado de interés que los autores de RNA otorgan a las diferentes materias. A continuación se presentan las ocho categorías temáticas utilizadas junto con una breve descripción de su contenido:

- Algoritmos: presentación de nuevos esquemas de aprendizaje o algoritmos, análisis de su rendimiento y presentación de métodos de optimización.

- Aplicaciones: aplicación práctica de las RNA en algún área de conocimiento: psicología, medicina, ingeniería, economía, etc.

- Comparación con otros modelos: comparación del rendimiento de las RNA con modelos estadísticos clásicos y modelos derivados de la Inteligencia Artificial.

- Epistemología: discusiones sobre filosofía de la mente y conexionismo.

- Hardware/Software: implementación en hardware de arquitecturas neuronales y presentación o evaluación de programas simuladores de RNA.

- Manual/Introducción: manuales de consulta y trabajos divulgativos o de introducción al campo de las RNA.

- Modelado de procesos: utilización de modelos conexionistas para el estudio y simulación de procesos fisiológicos (principalmente cerebrales) y cognitivos.

- Otros: trabajos muy generales sin ubicación específica.

Como se puede observar en la Figura 2, el área temática que cuenta con más registros es el modelado de procesos (2.139 registros), principalmente fisiológicos y psicológicos, perfilándose como la línea de investigación predominante. También acumulan un elevado número de publicaciones las categorías: aplicaciones (2.057 registros), manual/introducciones (1.665 registros) y algoritmos (1.024 registros).

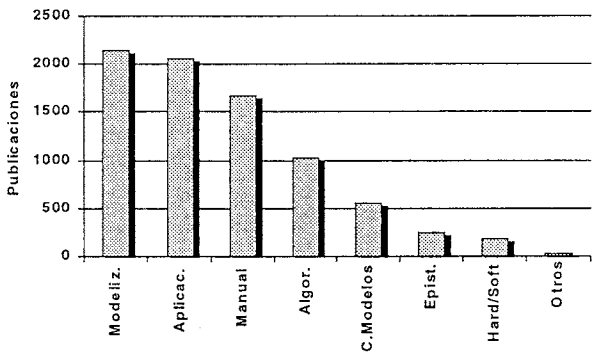


Figura 2. Distribución de las materias.

Aplicaciones más frecuentes

Para discriminar el tipo de aplicación de las RNA que se realiza mayoritariamente, los 2.057 registros se clasificaron en una de 43 áreas o disciplinas de aplicación, siendo las más frecuentes y por este orden: medicina (637 registros), ingeniería (597 registros), biología (362 registros) y psicología (132 registros) (ver Figura 3).

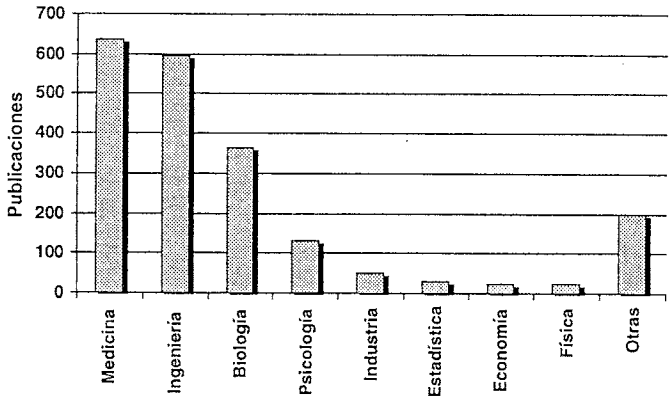


Figura 3. Áreas de aplicación más frecuentes.

Las áreas de aplicación abarcaban prácticamente cualquier disciplina de conocimiento (alimentación, aviación, agricultura, arqueología, documentación, hidrología, medio ambiente, música, tráfico, veterinaria, etc.).

Uso de las RNA en las diferentes áreas de la Psicología

Como grupo de investigación de las RNA en Psicología, examinamos mediante un análisis de contenido el papel que desempeñan las RNA en las diferentes áreas de nuestra disciplina. Para ello, nos centramos en el estudio de los 132 registros que trataban sobre la aplicación de RNA en este ámbito. Así, observamos que en el área de Evaluación, Personalidad y Tratamiento los autores se interesan principalmente por el diagnóstico de trastornos mentales (Pitarque, Ruiz, Fuentes, Martínez y García-Merita, 1997). Un ejemplo ilustrativo lo ofrece el trabajo de Zou, Shen, Shu, Wang, Feng et al. (1996), quienes desarrollaron una RNA con el objeto de clasificar un grupo de sujetos en una de tres categorías diagnósticas (neurosis, esquizofrenia o normal) a partir de las respuestas dadas a un cuestionario. El modelo resultante fue capaz de clasificar correctamente más del 90% de los sujetos.

En el área de Metodología los temas prioritarios versan sobre la clasificación de patrones y la aproximación de funciones. En este sentido, Pitarque, Roy y Ruiz (1998) han realizado recientemente una excelente comparación entre RNA y modelos estadísticos clásicos orientados a la clasificación y la predicción de valores. En este trabajo se ponen de manifiesto las cualidades de los modelos de red neuronal frente a los modelos estadísticos.

Por su parte, el área de Procesos Psicológicos Básicos está centrada en el modelado de procesos psicológicos y psicofísicos. Por ejemplo, MacWhinney (1998) se ha centrado en el desarrollo de modelos de adquisición del lenguaje mediante redes neuronales.

Los temas más recurrentes en el área de la Psicología Evolutiva tratan sobre la predicción del rendimiento académico (Hardgrave, Wilson y Walstrom, 1994) y la aplicación de modelos conexionistas en educación. En este sentido, Reason (1998) ha hecho uso de modelos PDP para crear programas de enseñanza de la lectura y para entender mejor por qué se producen dificultades de lectura en niños.

En el área de Psicología Social se trata generalmente de predecir y modelar di-

ferentes conductas sociales como, por ejemplo, el conocido dilema del prisionero (Macy, 1996).

Por último, los autores del área de psicofisiología se centran en el modelado de procesos psicofisiológicos (Olson y Grossberg, 1998) y en la clasificación de patrones EEG (Grözinger, Kögel y Röschke, 1998). Uno de los autores más prolíficos en esta última línea de investigación es Klöppel (1994).

Análisis de los estudios comparativos entre modelos estadísticos y RNA

Por último, de los 549 registros cuya área temática era la comparación entre RNA y otro tipo de modelos (estadísticos, sistemas expertos, etc.), nos centramos en el análisis de los 380 estudios que comparan de forma específica modelos estadísticos y RNA. Estos estudios fueron clasificados en función del objetivo que perseguían:

- Clasificación: asignación de la categoría de pertenencia de un determinado patrón y agrupamiento de patrones en función de las características comunes observadas entre los mismos.
- Predicción: estimación de variables cuantitativas
- Exploración/reducción: identificación de factores latentes y reducción de espacios de alta dimensión.

Los resultados reflejan que los trabajos cuyo objetivo es la clasificación representan el 71% de este tipo de estudios comparativos. En este sentido, los modelos más frecuentemente comparados son el análisis discriminante, la regresión logística y el análisis de clusters. Los trabajos más sobresalientes y que constituyen puntos de referencia son los de Balakrishnan, Cooper, Jacob y Lewis (1994), Michie, Spiegelhalter y Taylor (1994) y, más recientemente, Waller, Kaiser, Illian y Manry (1998). Los trabajos sobre predicción representan el 27% del total de estudios comparativos, siendo los modelos estadísticos más frecuentes la regresión lineal, el análisis de series temporales y los modelos de supervivencia. En este grupo de trabajos podemos destacar los de Ohno-Machado (1997), Pitarque, Roy y Ruiz (1998) y Tsui (1996).

Por último, el modelo estadístico generalmente utilizado en los estudios de exploración/reducción es el análisis de componentes principales, como en el trabajo de Garrido, Gaitan, Serra y Calbet (1995). Estos trabajos representaban únicamente el 2% del total.

De los resultados comentados se deduce claramente que los estudios comparativos de predicción y exploración/reducción representan líneas de investigación minoritarias respecto a los estudios comparativos de clasificación.

Finalmente, respecto a los resultados que se obtienen en el conjunto de los estudios comparativos revisados, aproximadamente el 80% concluye a favor de un mejor rendimiento de las RNA frente a los modelos estadísticos clásicos. Sin embargo, este dato debe ser interpretado con cierta precaución, ya que en varios de estos trabajos hemos detectado algunas deficiencias como, por ejemplo, la utilización de muestras reducidas - muchas de ellas de tipo clínico -, la falta de información acerca del cumplimiento de supuestos y el uso de pruebas de evaluación de rendimiento sesgadas o inadecuadas.

Conclusiones

El estudio realizado ha permitido al Grupo ERNAP documentarse en las diferentes líneas de investigación prioritarias en el ámbito de las RNA y, a partir de ellas, hemos iniciado cuatro líneas de investigación.

La primera trata sobre la comparación entre modelos estadísticos de clasificación y RNA que, como se ha visto, es un tema predominante dentro de los estudios comparativos (Navarro y Losilla, 2000). En segundo lugar, nos planteamos abordar el análisis de la supervivencia mediante RNA; se trata ésta de una línea de investigación incipiente y en la que, por tanto, se hace necesaria la aportación de datos tanto teóricos como empíricos. En tercer lugar, hemos llevado a cabo la aplicación de RNA en el análisis de datos con información faltante, mostrando las RNA un rendimiento superior frente a los procedimientos clásicos (Navarro, 1998). Por último, en consonancia con una de las líneas de investigación más prioritarias, hemos analizado las condiciones necesarias para el modelado de los procesos psicológicos mediante RNA.

Aunque los estudios e investigaciones realizados por nuestro grupo son preliminares, los resultados pueden ser calificados como de muy satisfactorios, constatando la utilidad de las RNA en el campo de la psicología y de la metodología.

Referencias

Alcain, M.D. (1991). Aspectos métricos de la información científica. *Ciencias de la Información (La Habana),* diciembre, 32-36.

Amat, N. (1994). *La documentación y sus tecnologías.* Madrid: Pirámide.

Anderson, J., Silverstein, J., Ritz, S. y Jones, R. (1977). Distinctive features, categorical perception and probability learning: some applications on a neural model. Psychological Review, 84, 413-451.

Bainbridge, W.S. (1995). Neural network models of religious belief. *Sociological Perspectives, 38(4),* 483-495.

Balakrishnan, P.V., Cooper, M.C., Jacob, V.S. y Lewis, P.A. (1994). A study ot the classification capabilities of neural networks using unsupervised learning: a comparison with k-means clustering. *Psychometrika, 59(4),* 509-525.

Carpenter, G. y Grossberg, S. (1986). *Absolutely stable learning of recognition codes by a self-organizing neural network.* American Institute of Physics (AIP) Conference Proceedings 151: Neural Networks for Computing, 77-85.

Carpintero, H. (1980). La psicología actual desde una perspectiva bibliométrica: Una introducción. *Análisis y Modificación de Conducta, 6(11-12),* 9-23.

Caudill, M. y Butler, C. (1992). *Understanding neural networks: Computer explorations.* Cambridge, MA: MIT Press.

Ferreiro, L. (1993). Bibliometría (Análisis Bivariante). Madrid: EYPASA.

Fukushima, K. (1988). Neocognitron: A hierarchical neural network model capable of visual pattern recognition. *Neural Networks, 1(2),* 119-130.

Garrido, L., Gaitan, V., Serra, R.M. y Calbet, X. (1995). Use of multilayer feedforward neural nets as a display method for multidimensional distributions. *International Journal of Neural Systems, 6(3),* 273-82.

Grözinger, M., Kögel, P. y Röschke, J. (1998). Effects of Lorazepam on the automatic online evaluation of sleep EEG data in healthy volunteers. *Pharmacopsychiatry, 31(2),* 55-59.

Hardgrave, B.C., Wilson, R.L. y Walstrom, K.A. (1994). Predicting graduate student success: A comparison of neural networks and traditional techniques. *Computers Operation Research, 21(3),* 249-263.

Hopfield, J. (1982). Neural networks and physical systems with emergent colective computational abilities. *Proceedings of the National Academy of Sciences, 79,* 2554-2558.

Hopfield, J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, 81,* 3088-3092.

Klöppel, B. (1994). Neural networks as a new method for EEG analysis: A basic introduction. *Neuropsychobiology, 29,* 33-38.

Kohonen, T. (1984). *Self-Organization and associative memory.* Berlin: Springer-Verlag.

Losilla, J.M. (1997). *Ebla. Gestor bibliográfico: Manual de uso.* Barcelona: Signo.

MacWhinney, B. (1998). Models of the emergence of language. *Annual Review of Psychology, 49,* 199-227.

Macy, M. (1996). Natural Selection and Social Learning in Prisoner's Dilemma: Coadaptation with Genetic Algorithms and Artificial Neural Networks. *Sociological Methods and Research, 25(1),* 103-137.

Méndez, A. (1986). Los indicadores bibliométricos. *Política Científica, Octubre, 34-36.*

Michie, D., Spiegelhalter, D.J. y Taylor, C.C. (1994). *Machine learning, neural and statistical classification.* New York: Ellis Horwood.

Navarro, J.B. (1998). *Aplicación de redes neuronales artificiales al tratamiento de datos incompletos.* Tesis doctoral publicada en microficha. Bellaterra, Barcelona: Servei de Publicacions. Universitat Autónoma de Barcelona.

Navarro, J.B. y Losilla, J.M. (2000). Análisis de datos faltantes mediante redes neuronales artificiales: Un estudio de simulación. *Psicothema, 12,502-510.*

Ohno-Machado, L. (1997). A comparison of Cox proportional hazards and artificial neural network models for medical prognosis. *Computational Biology in Medicine, 27(1),* 55-65.

Olson, S.J. y Grossberg, S. (1998). A neural network model for the development of simple and complex cell receptive fields within cortical maps of orientation and ocular dominance. *Neural Networks, 11(2),* 189-208.

Pitarque, A., Roy, J.F. y Ruiz, J.C. (1998). Redes neurales vs. modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación. *Psicológica, 19,* 387-400.

Pitarque, A., Ruiz, J.C., Fuentes, I., Martínez, M.J. y García-Merita, M. (1997). Diagnóstico clínico en psicología a través de redes neurales. *Psicothema, 9(2),* 359-363.

Radin, D.I. y Rebman, J.M. (1998). Seeking psi in the casino. *Journal of the Society for Psychical Research, 62(850),* 193-219.

Reason, R. (1998). How relevant is connectionist modelling of reading to educational practice? Some implications of Margaret Snowling's article. *Educational and Child Psychology, 15(2),* 59-65. item Romera, M.J. (1992). Potencialidad de la bibliometría para el estudio de la ciencia. Aplicación a la educación especial. *Revista de Educación, 297,* 459-478.

Rosenblatt, F. (1958). The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65,* 386-408.

Rumelhart, D.E., Hinton, G.E. y McClelland, J.L. (1986). A general framework for parallel distributed processing. En: D.E. Rumelhart, J.L. McClelland y el grupo PDP (Eds.). *Parallel distributed processing* (pp. 45-76). Cambridge, MA: MIT Press.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart, J.L. McClelland y el grupo PDP (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rumelhart, D.E., McClelland, J.L. y el grupo PDP (Eds.) (1986). *Parallel distributed processing: Explorations in the microstrucure of cognition. Vol. 1: Foundations.* Cambridge, MA: MIT Press.

Sancho, R. (1990). Indicadores bibliométricos utilizados en la evaluación de la ciencia y la tecnología. Revisión bibliográfica. *Revista Española de Documentación Científica, 13(3-4),* 842-865.

Tsui, F.C. (1996). Time series prediction using a multiresolution dynamic predictor: Neural network (University of Pittsburgh, 1996). *Dissertation Abstracts International, DAI-B 58/03,* 1456.

Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psychometrika, 63(1),* 5-22.

Zou, Y., Shen, Y., Shu, L., Wang, Y., Feng, F., Xu, K., Qu, Y., Song, Y., Zhong, Y., Wang, M. y Liu, W. (1996). Artificial neural network to assist psychiatric diagnosis. *British Journal of Psychiatry, 169,* 64-67.

Original recibido: 24/03/2000
Versión final aceptada: 12/09/2000

2.4.
Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo.

Psicothema
2002. Vol. 14, nº 3, pp. 630-636

ISSN 0214 - 9915 CODEN PSOTEG

Redes neuronales artificiales aplicadas al análisis de supervivencia: un estudio comparativo con el modelo de regresión de Cox en su aspecto predictivo

Alfonso Palmer Pol y Juan José Montaño Moreno
Universidad de las Islas Baleares

El objetivo de este estudio fue comparar el rendimiento en predicción entre los modelos de Redes Neuronales Artificiales (RNA) y el modelo de riesgos proporcionales de Cox en el contexto del análisis de supervivencia. Más concretamente, se intentó comprobar: a) si el modelo de redes neuronales jerárquicas es más preciso que el modelo de Cox, y b) si el modelo de redes neuronales secuenciales supone una mejora respecto al modelo de redes neuronales jerárquicas. La precisión fue evaluada a partir de medidas de resolución (área bajo la curva ROC) y calibración (prueba de Hosmer-Lemeshow) usando un conjunto de datos de supervivencia. Los resultados mostraron que las redes neuronales jerárquicas tienen un mejor rendimiento en resolución que el modelo de Cox, mientras que las redes secuenciales no suponen una mejora respecto a las redes neuronales jerárquicas. Finalmente, los modelos de RNA proporcionan curvas de supervivencia más ajustadas a la realidad que el modelo de Cox.

Artificial neural networks applied to the survival analysis: A comparative study with Cox regression model in its predictive aspect. The purpose of this study was to compare the performance in prediction between the models of Artificial Neural Networks (ANN) and Cox proportional hazards models in the context of survival analysis. More specifically, we tried to verify: a) if the model of hierarchical neural networks is more accurate than Cox's model, and b) if the model of sequential neural networks signifies an improvement with respect to the hierarchical neural networks model. The accuracy was evaluated through resolution (the area under the ROC curve) and calibration (Hosmer-Lemeshow test) measures using survival data. Results showed that hierarchical neural networks outperform Cox's model in resolution while sequential neural networks do not suppose an improvement with respect to hierarchical neural networks. Finally, ANN models produced survival curves that were better adjusted to reality than Cox's model.

La presencia de información incompleta o censurada constituye una característica fundamental en los datos de supervivencia que hace difícil su manejo mediante los métodos estadísticos convencionales (Allison, 1995). En este tipo de datos también se pueden utilizar variables dependientes del tiempo, esto es, variables cuyos valores pueden cambiar a lo largo del período de observación.

El modelo de regresión de riesgos proporcionales, conocido habitualmente como modelo de regresión de Cox (Cox, 1972), es el modelo más utilizado en este contexto y relaciona la función de riesgo con las variables explicativas por medio de la expresión:

$$h(t, \boldsymbol{X}) = h_0(t)e^{\beta' \boldsymbol{X}}$$

Fecha recepción: 2-10-01 • Fecha aceptación: 22-1-02
Correspondencia: Alfonso Palmer Pol
Facultad de Psicología
Universidad de las Islas Baleares
07071 Palma de Mallorca (Spain)
E-mail: alfonso.palmer@uib.es

Un aspecto importante del modelo de Cox radica en que éste se puede utilizar para realizar predicciones sobre el proceso de cambio. Más concretamente, en el presente trabajo, nos proponemos utilizar el modelo de Cox para predecir la función de supervivencia, con unos determinados valores en las variables explicativas. La función de supervivencia para un sujeto dado se puede obtener mediante el modelo de Cox a través de la siguiente expresión:

$$S(t, \boldsymbol{X}) = S_0(t)^{e^{\beta' \boldsymbol{X}}}$$

Para que una variable explicativa pueda formar parte de este modelo se debe verificar si ésta cumple el «supuesto de proporcionalidad» (Allison, 1984). En el caso que se incumpla este supuesto, habitualmente se excluye del modelo la variable explicativa y ésta se trata como variable de estrato (Blossfeld y Rohwer, 1995; Marubini y Valsecchi, 1995; Parmar y Machin, 1995).

La utilización de las Redes Neuronales Artificiales (RNA) se ha centrado principalmente en la clasificación de patrones y en la estimación de variables cuantitativas, sin embargo, apenas existen aplicaciones en el campo del análisis de supervivencia. En este sentido, podemos considerar pioneros los trabajos de Ohno-Ma-

chado (Ohno-Machado, Walker y Musen, 1995; Ohno-Machado y Musen, 1997a; Ohno-Machado y Musen, 1997b), quien ha propuesto dos modelos de red neuronal que permiten el manejo de datos de supervivencia sin necesidad de imponer ningún supuesto de partida, susceptibles de ser un buen complemento al modelo de Cox: el modelo de redes jerárquicas y el modelo de redes secuenciales.

El modelo de redes jerárquicas (Ohno-Machado, Walker y Musen, 1995) consiste en una arquitectura jerárquica de redes neuronales del tipo perceptrón multicapa que predicen la supervivencia mediante un método paso a paso (ver figura 1). De este modo, cada red neuronal se encarga de dar como salida la probabilidad de supervivencia en un intervalo de tiempo determinado, proporcionando el modelo general la supervivencia para el primer intervalo, después para el segundo intervalo y así sucesivamente.

El modelo de redes secuenciales (Ohno-Machado y Musen, 1997a; Ohno-Machado y Musen, 1997b) supone una ampliación respecto al modelo de redes jerárquicas. En el modelo de redes secuenciales la predicción realizada por una red neuronal para un intervalo de tiempo puede actuar a su vez como variable explicativa o de entrada en otra red dedicada a la predicción de otro intervalo anterior o posterior (ver figura 2). Con esta estrategia, se pretende modelar explícitamente la dependencia temporal que existe entre las predicciones realizadas en los diferentes intervalos de tiempo y así obtener curvas de supervivencia asintóticamente decrecientes.

Con el presente estudio se pretende comprobar, por un lado, si el modelo de redes jerárquicas presenta un rendimiento superior en cuanto a predicción frente al modelo de regresión de Cox y, por otro lado, si el modelo de redes secuenciales supone una mejora en rendimiento respecto al modelo de redes jerárquicas. Estas dos hipótesis serán contrastadas a partir de un conjunto de datos de supervivencia derivado del campo de las conductas adictivas.

Materiales y métodos

Matriz de datos

Los datos utilizados en la presente investigación proceden de una serie de estudios realizados por el equipo de McCusker (McCusker et al., 1995; McCusker, Bigelow, Frost et al., 1997; McCusker, Bigelow, Vickers-Lahti, Spotts, Garfield y Frost, 1997) en la Universidad de Massachusetts (la matriz de datos, denominada uis.dat, se puede obtener en la sección *Survival Analysis* de la siguiente dirección URL: http://www-unix.oit.umass.edu/~statdata). El objetivo de estos estudios fue comparar diferentes programas de intervención diseñados para la reducción del abuso de drogas en una muestra de 628 toxicómanos. Estos programas podían diferir en función de la duración de la intervención (corta o larga) y de la orientación terapéutica (clínica A o clínica B). En la tabla 1 se presenta la descripción de las nueve variables explicativas utilizadas en la investigación. No se utilizaron variables dependientes del tiempo. La variable de respuesta fue el tiempo en días transcurrido desde el inicio del estudio hasta la recaída del sujeto en el consumo de drogas. Por tanto, el suceso de interés fue el cambio de estado de no consumo a consumo de drogas. El seguimiento de los sujetos se realizó a lo largo de tres años y medio.

En el gráfico 1 se puede observar la representación gráfica del estimador Kaplan-Meier de la función de supervivencia sobre los datos del estudio.

A partir del valor de los deciles obtenidos mediante Kaplan-Meier, se determinaron diferentes intervalos de tiempo. El decil 9 no fue utilizado debido a que los valores censurados se acumulan al final del seguimiento, como puede observarse en el gráfico, y apenas hay cambios de estado en ese período. De esta forma, se obtuvieron ocho intervalos de tiempo en los que la probabilidad de supervivencia se va decrementando de forma aproximadamente

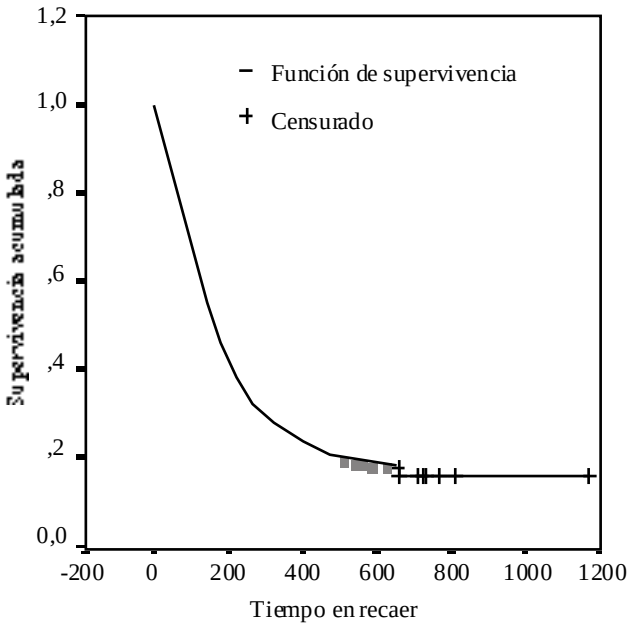


Gráfico 1. *Función de supervivencia estimada mediante el método Kaplan-Meier*

Tabla 1
Descripción de las variables explicativas

Variable	Descripción	Valores
Edad	Edad al entrar en el estudio	años
Beck	Puntuación en el Inventario de Depresión de Beck	0 - 54
Hercoc	Uso de heroína/cocaina 3 meses antes de la admisión	1= heroína y cocaina; 2= solo heroína; 3= solo cocaina; 4= ni heroína, ni cocaina
Hdroga	Historia del uso de droga	1= nunca; 2= anterior; 3= reciente
Ntrat	Número de tratamientos previos contra la droga	0 - 40
Raza	Raza del sujeto	0= blanca; 1= no blanca
Trat	Asignación aleatoria de tratamiento	0= corto; 1= largo
Lugar	Lugar del tratamiento	0= clínica A; 1= clínica B
Durac	Duración de estancia en tratamiento (fecha de admisión a fecha de salida)	

constante a medida que avanza el seguimiento. A continuación, se procedió a dividir aleatoriamente la muestra total en dos grupos de forma que la proporción de cambio en cada intervalo era aproximadamente la misma en ambos grupos: 528 sujetos actuaron como grupo de entrenamiento para la construcción de los modelos y 100 sujetos actuaron como grupo de test para la comparación entre modelos. En la tabla 2 se muestra, para cada intervalo de tiempo considerado, los días de seguimiento que comprende el intervalo, la distribución acumulada de cambios y no cambios de estado y la proporción acumulada de cambio, para el grupo de entrenamiento y el de test.

Modelo de Cox

Para la generación del modelo de Cox se procedió, en primer lugar, a comprobar el supuesto de proporcionalidad en las nueve variables explicativas. Se pudo observar que la variable «duración de estancia en tratamiento» (Durac) no cumple el supuesto de proporcionalidad y, en consecuencia, quedó excluida del modelo para ser utilizada como variable de estratificación con dos estratos. A continuación, fueron introducidas en el modelo las ocho variables explicativas restantes y todos los términos de interacción de primer orden (los términos de interacción de segundo orden y de orden superior no fueron introducidos, debido a que el método de estimación de los parámetros del modelo no alcanzaba la convergencia). Se generaron variables ficticias para aquellas variables nominales con más de dos categorías. Mediante un método de selección paso a paso hacia atrás basado en la razón de verosimilitud (*backward stepwise: likelihood ratio*), quedaron incluidas en el modelo cuatro variables y siete términos de interacción.

Modelos de redes neuronales

El modelo de redes neuronales jerárquicas estaba compuesto por ocho redes del tipo perceptrón multicapa como las presentadas en la figura 1, cada una estaba centrada en dar como salida la probabilidad de supervivencia en uno de los intervalos de tiempo creados.

A fin de obtener el modelo de red óptimo en cuanto a predicción y evitar así el fenómeno del sobreajuste, se utilizó un grupo de 100 sujetos de entrenamiento seleccionado aleatoriamente como grupo de validación. La configuración neuronal que exhibiera el mejor rendimiento ante el grupo de validación sería el modelo seleccionado para pasar a la fase de test. Se probaron diferentes arquitecturas en cuanto al número de neuronas en la capa oculta, funciones de activación de las neuronas y algoritmos de aprendizaje como el *backpropagation* (Rumelhart, Hinton y Williams, 1986) y alguna de sus variantes más utilizadas: *quickpropagation* (Fahlman, 1988), *delta-bar-delta* (Jacobs, 1988), gradiente conjugado (Battiti, 1992), *resilient propagation* (Smith, 1993). Finalmente, se utilizaron ocho redes perceptrón multicapa con dos neuronas en la capa oculta, función de activación tangente hiperbólica en la capa oculta y lineal en la capa de salida, y entrenadas mediante el algoritmo de gradiente conjugado.

En el modelo de redes secuenciales la predicción realizada por una red neuronal para un intervalo de tiempo actúa a su vez como variable explicativa o de entrada en otra red dedicada a la predicción de otro intervalo anterior o posterior. De esta forma, el intervalo correspondiente a la primera red neuronal actúa como intervalo informativo y el intervalo correspondiente a la segunda red neuronal actúa como intervalo informado (figura 2). Siguiendo este esquema se cruzaron los ocho modelos jerárquicos correspondientes a los ocho intervalos de tiempo, generándose 56 redes secuenciales.

Para la generación de las arquitecturas neuronales se empleó el programa *Neural Connection 2.1* (SPSS Inc., 1998) que permite simular el comportamiento de una red perceptrón multicapa asociada al algoritmo de gradiente conjugado.

Técnicas de comparación

Se comparó la eficacia de los modelos presentados a partir de las predicciones realizadas sobre los 100 sujetos de test. La eficacia en cuanto a predicción se determinó a partir de medidas de resolución y calibración.

La resolución hace referencia a la capacidad de discriminar por parte del modelo entre sujetos que realizan el cambio de estado y sujetos que no realizan el cambio. La resolución se midió a partir del área bajo la curva ROC (*Receiver Operating Characteristics*) (Swets, 1973, 1988). La comparación entre dos áreas bajo la curva ROC se realizó mediante la prueba z descrita por Hanley y McNeil (1983).

La calibración hace referencia a lo cerca que se encuentran las probabilidades proporcionadas por el modelo respecto al resultado real. La calibración se midió a partir de la prueba χ^2 de Hosmer-Lemeshow (Hosmer y Lemeshow, 1980).

Resultados

Comparación modelo de Cox versus modelo de redes jerárquicas

En relación a la comparación en función de la resolución, el gráfico 2 muestra las áreas bajo la curva ROC del modelo de Cox

Tabla 2
Distribución acumulada de casos en grupo de entrenamiento y test de acuerdo al intervalo de tiempo

	Grupo de Entrenamiento				Grupo de Test			
Intervalo	**Cambio**	**No Cambio**	**Total**	**Propor.**	**Cambio**	**No Cambio**	**Total**	**Propor.**
1º 1-26	53	475	528	0.10	10	90	100	0.10
2º 27-59	106	422	528	0.20	20	80	100	0.20
3º 60-90	161	367	528	0.30	30	70	100	0.30
4º 91-120	212	316	528	0.40	39	61	100	0.39
5º 121-166	265	263	528	0.50	50	50	100	0.50
6º 167-220	320	208	528	0.60	60	40	100	0.60
7º 221-290	371	157	528	0.70	70	30	100	0.70
8º 291-501	423	104	527	0.80	79	21	100	0.79

y el modelo de redes jerárquicas. Ambos modelos presentan una buena precisión diagnóstica, excepto en el caso del modelo de Cox para el intervalo número ocho que proporciona un área bajo la curva ROC por debajo de 0.70 (Swets, 1988). Se puede observar que el modelo de redes exhibe un rendimiento superior en todos los intervalos considerados. Esta superioridad se comprueba a nivel estadístico mediante la prueba z (con riesgo unilateral) (Hanley y McNeil, 1983). Por otra parte, se observa que el error estándar del

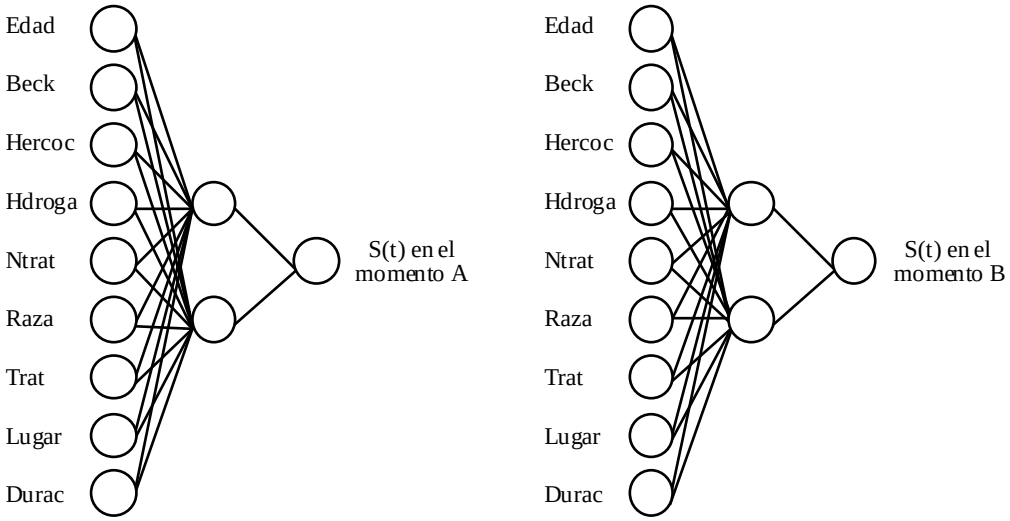


Figura 1. *Modelo de red jerárquica*

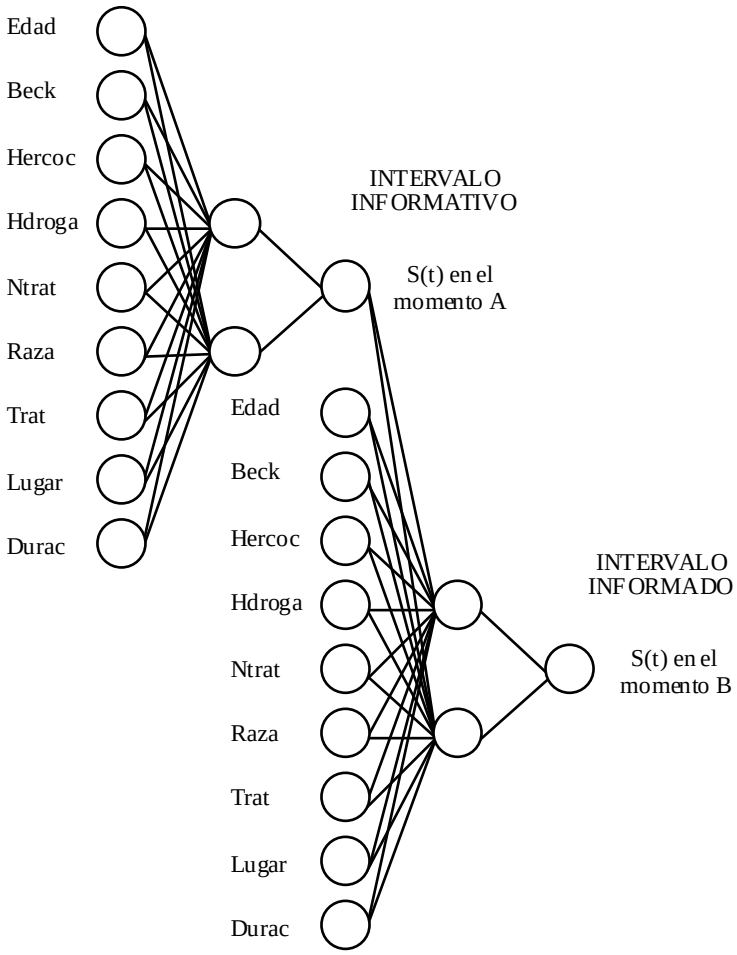


Figura 2. *Modelo de red secuencial*

área bajo la curva ROC obtenido con el modelo de redes es sistemáticamente inferior que el obtenido con el modelo de Cox. Este hecho implica que las estimaciones del área bajo la curva ROC del modelo de redes tendrán más precisión que en el caso del modelo de Cox.

En relación a la calibración, en la tabla 3 se puede observar a través de la prueba de Hosmer-Lemeshow (1980) que ambos modelos tienen un buen ajuste en todos los intervalos de tiempo considerados, debido a que la discrepancia entre lo observado y lo esperado no es significativa. En este caso, no se aprecian diferencias importantes entre el modelo de Cox y el modelo de redes jerárquicas respecto a la medida de calibración.

Comparación modelo de redes jerárquicas versus modelo de redes secuenciales

En relación a la comparación en función de la resolución, la tabla 4 muestra los resultados de la prueba z (Hanley y McNeil, 1983) que permite comparar el área bajo la curva ROC del modelo de redes jerárquicas y del modelo de redes secuenciales. Los valores z positivos indican un mejor rendimiento por parte del modelo de redes secuenciales, mientras que los valores z negativos indican un mejor rendimiento por parte del modelo de redes jerárquicas. En ningún caso, las redes secuenciales mostraron un rendimiento significativamente superior frente a la versión jerárquica. Más bien, se puede observar que el rendimiento de las redes secuenciales fue inferior en numerosos casos. En este sentido, el ejemplo más significativo es la red que predice la probabilidad de supervivencia en el intervalo número cinco utilizando como intervalo informativo el número tres (z= -4.76, p<.01).

En relación a la calibración, la tabla 5 muestra los resultados de la prueba de Hosmer-Lemeshow (1980) para las 56 redes secuenciales. Todos las redes secuenciales mostraron un buen ajuste, aunque no supuso una mejora en rendimiento respecto al modelo de redes jerárquicas (ver tabla 5). Se puede apreciar que, en general, el rendimiento mejoraba cuando se utilizaba como intervalo informativo el intervalo inmediatamente posterior.

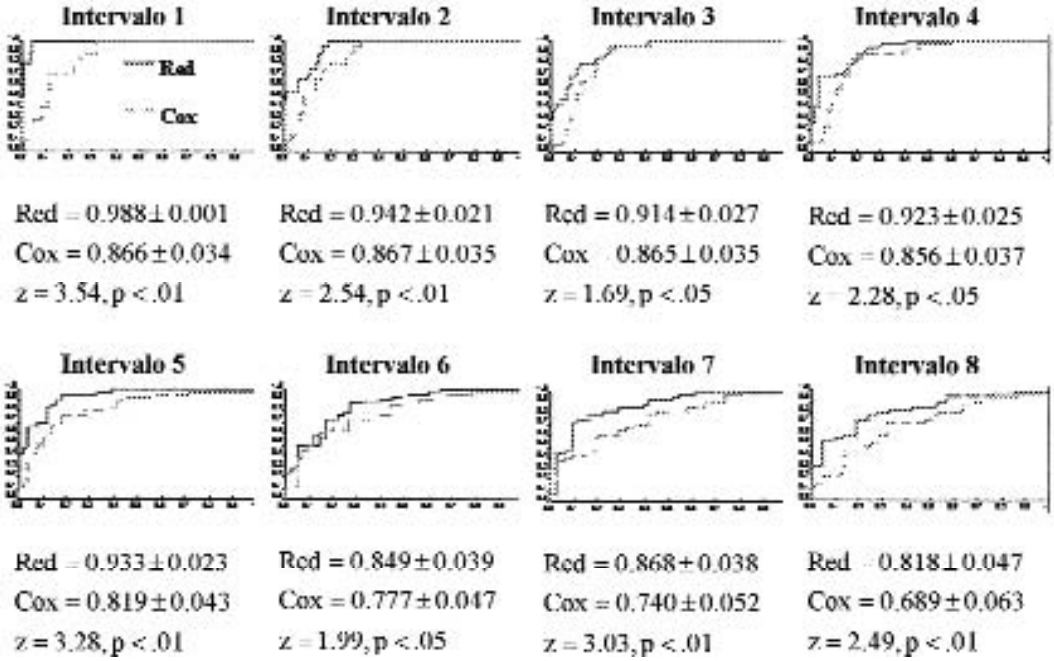


Gráfico 2. *Áreas bajo la curva ROC del modelo de redes jerárquicas y el modelo de Cox*

Tabla 3
Calibración del modelo de Cox y el modelo de redes jerárquicas

	Prueba χ^2	
Intervalo	**Modelo de Cox**	**Redes jerárquicas**
1	0.654	1.516
2	1.053	1.928
3	4.373	1.941
4	4.067	4.786
5	4.632	4.469
6	5.574	5.393
7	5.334	5.270
8	5.112	6.050

Con el objeto de demostrar que los modelos de RNA analizados también pueden servir para generar curvas de supervivencia ajustadas tanto para sujetos como para grupos, se muestran en el gráfico 3 las curvas de supervivencia estimadas por los tres modelos –modelo de Cox, modelo de redes jerárquicas y modelo de redes secuenciales generado mediante la utilización del intervalo posterior a cada momento como intervalo informativo– para un sujeto perteneciente al grupo de test que realizó el cambio de estado en el intervalo número cinco. En la gráfica se aprecia cómo los modelos de redes se ajustan más a la realidad que el modelo de Cox. Más concretamente, para el intervalo en que se produce el cambio, las redes jerárquicas y secuenciales proporcionan una es-

Tabla 4
Comparación en resolución entre el modelo de redes jerárquicas y el modelo de redes secuenciales

Prueba z	**Intervalo informado**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Intervalo informativo								
1		-0.66	0.06	-0.21	-0.07	1.05	-0.41	0.25
2	-0.21		-1.03	0.15	0.01	0.77	0.06	0.33
3	0.11	0.36		-0.32	-4.76**	0.66	-0.46	0.24
4	0.00	0.00	0.70		0.46	1.45	-0.35	-0.78
5	0.00	-0.15	-0.30	0.17		1.08	0.03	0.46
6	0.13	-0.04	0.20	-0.65	-1.04		-0.51	-0.12
7	0.00	0.20	0.46	0.27	-0.38	0.72		-0.10
8	0.13	0.41	0.02	0.05	-0.01	1.12	-0.12	

Nota: ** p <.01 prueba unilateral

Tabla 5
Calibración del modelo de redes secuenciales

Prueba χ^2	**Intervalo informado**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Intervalo informativo								
1		1.620	1.216	0.952	5.281	3.196	7.305	5.982
2	0.859		0.241	2.883	5.799	5.004	3.407	3.544
3	2.478	1.196		2.100	4.164	6.551	9.204	9.404
4	1.570	1.932	0.972		5.075	2.044	5.011	7.035
5	1.484	1.929	1.641	2.950		7.071	5.072	8.068
6	1.646	1.806	0.620	2.018	3.151		2.723	1.897
7	1.668	1.658	1.405	3.959	4.489	1.343		7.793
8	1.481	2.419	0.812	2.581	4.356	4.015	6.767	

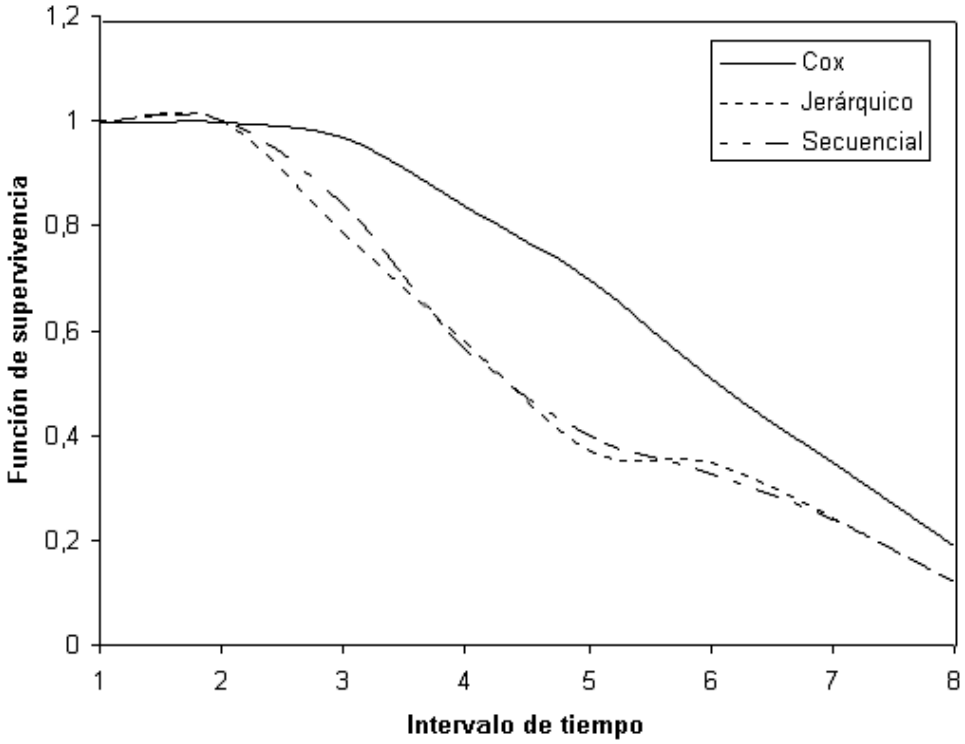


Gráfico 3. *Estimación de la función de supervivencia de los tres modelos analizados para un sujeto de test que realizó el cambio en el intervalo 5*

timación de la función de supervivencia de 0.3694 y 0.4005, respectivamente; mientras que para el modelo de Cox esta estimación es de 0.6947. Por otra parte, se observa que el modelo de redes secuenciales suaviza ligeramente la curva respecto al modelo de redes jerárquicas. Aunque en algunos casos se han observado pequeñas anomalías en las curvas de supervivencia obtenidas mediante los modelos de redes, para la mayoría de individuos estas curvas son monótonamente decrecientes y, como en el ejemplo comentado, más ajustadas a la realidad que el modelo de Cox.

Conclusiones

En el presente estudio se ha comprobado que los modelos de RNA jerárquicos y secuenciales permiten el manejo de datos de supervivencia sin necesidad de imponer supuestos de partida en los datos. La información parcial proporcionada por los datos censurados es utilizada en aquellas redes neuronales para las que se tiene información del cambio de estado en el intervalo de tiempo correspondiente. Por ejemplo, los datos de un sujeto al que se le haya realizado el seguimiento hasta el tercer intervalo considerado serán usados en las redes correspondientes al primer, segundo y tercer intervalo, pero no en las redes correspondientes a los siguientes intervalos de tiempo. Si bien en este trabajo no se han utilizado variables dependientes del tiempo, éstas se pueden incorporar fácilmente debido a que cada red neuronal puede recibir, en cada momento temporal, un valor diferente respecto a las variables explicativas para un mismo sujeto.

La comparación llevada a cabo en cuanto a poder predictivo entre los modelos presentados ha puesto de manifiesto, en primer lugar, que el modelo de redes secuenciales tiene una resolución significativamente mejor que el modelo de Cox, mientras que ambos modelos han mostrado una calibración similar. En segundo lugar, el modelo de redes secuenciales no ha supuesto una mejora en rendimiento respecto al modelo de redes jerárquicas, observándose en algunos casos una peor ejecución por parte del primer modelo. Así pues, con los datos manejados en este estudio, no se obtienen las ventajas descritas por Ohno-Machado (1996) en cuanto a las redes secuenciales. En el trabajo de Ohno-Machado (1996), las redes jerárquicas mostraron un rendimiento superior tanto en resolución como calibración frente al modelo de Cox en la mayoría de intervalos de tiempo considerados. Por su parte, las redes secuenciales no mostraron un mejor rendimiento en calibración respecto al modelo de redes jerárquicas, al igual que en el presente trabajo, aunque en la mayoría de intervalos de tiempo sí obtuvieron una mejor resolución.

Por último, se ha comprobado la utilidad de los modelos de red para realizar curvas de supervivencia tanto individuales como grupales, exhibiendo éstas un mejor ajuste en la estimación de la función de supervivencia frente al modelo de Cox.

Este conjunto de resultados pone de manifiesto que las RNA pueden ser útiles en el análisis de datos de supervivencia.

Referencias

Allison, P.D. (1984). *Event history analysis. Regression for longitudinal event data*. Beverly Hills, CA: Sage Pub.

Allison, P.D. (1995). *Survival analysis using the SAS system: a practical guide*. Cary, NC: SAS Institute Inc.

Battiti, R. (1992). First and second order methods for learning: between steepest descent and Newton's method. *Neural Computation, 4*(2), 141-166.

Blossfeld, H.P. y Rohwer, G. (1995). *Techniques of event history modeling*. Mahwah, NJ: Lawrence Erlbaum Associates, Pub.

Cox, D.R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B*, 34, 187-202.

Fahlman, S.E. (1988). Faster-learning variations on back-propagation: an empirical study. En: D. Touretsky, G.E. Hinton y T.J. Sejnowski (Eds.). *Proceedings of the 1988 Connectionist Models Summer School* (pp. 38-51). San Mateo: Morgan Kaufmann.

Hanley, J.A. y McNeil, B.J. (1983). A method of comparing the areas under receiver operating characteristics curves derived from the same cases. *Radiology, 148*, 839-843.

Hosmer, D.W. y Lemeshow, S. (1980). A goodness-of-fit test for the multiple logistic regression model. *Communications in Statistics, A10*, 1.043-1.069.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*(4), 295-308.

Marubini, E. y Valsecchi, M.G. (1995). *Analysing survival data from clinical trials and observational studies*. New York: John Wiley and Sons.

McCusker, J., Bigelow, C., Frost, R., Garfield, F., Hindin, R., Vickers-Lahti, M. y Lewis, B.F. (1997). The effects of planned duration of residential drug abuse treatment on recovery and HIV risk behavior. *American Journal of Public Health, 87*, 1.637-1.644.

McCusker, J., Bigelow, C., Vickers-Lahti, M., Spotts, D., Garfield, F. y Frost, R. (1997). Planned duration of residential drug abuse treatment: efficacy versus treatment. *Addiction, 92*, 1.467-1.478.

McCusker, J., Vickers-Lahti, M., Stoddard, A.M., Hindin, R., Bigelow, C., Garfield, F., Frost, R., Love, C. y Lewis, B.F. (1995). The effectiveness of alternative planned durations of residential drug abuse treatment. *American Journal of Public Health, 85*, 1.426-1.429.

Ohno-Machado, L. (1996). *Medical applications of artificial neural networks: connectionist models of survival*. Tesis doctoral no publicada. Stanford University.

Ohno-Machado, L. y Musen, M. (1997a). Modular neural networks for medical prognosis: quantifying the benefits of combining neural networks for survival prediction. *Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research, 9*(1), 71-86.

Ohno-Machado, L. y Musen, M. (1997b). Sequential versus standard neural networks for pattern recognition: an example using the domain of coronary heart disease. *Computational Biology in Medicine, 27*(4), 267-281.

Ohno-Machado, L., Walker, M. y Musen, M. (1995). Hierarchical neural networks for survival analysis. *Medinfo, 8 Pt 1*, 828-832.

Parmar, M.K.B. y Machin, D. (1995). *Survival analysis: a practical approach*. New York: John Wiley and Sons.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.

SPSS Inc. (1998). *Neural Connection 2.1* [Programa para ordenador]. SPSS Inc. (Productor). Chicago: SPSS Inc. (Distribuidor).

Swets, J.A. (1973). The relative operating characteristic in psychology. *Science, 182*, 990-1.000.

Swets, J.A. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*, 1.285-1.293.

2.5.
Redes neuronales artificiales: abriendo la caja negra.

Metodología de las Ciencias del Comportamiento 4(1), 77–93

Redes neuronales artificiales: abriendo la caja negra

Juan José Montaño Moreno[1] Alfonso Palmer Pol
Universidad de las Islas Baleares
Carlos Fernández Provencio
InfoMallorca, S.L.

Resumen

El mayor esfuerzo en la investigación sobre Redes Neuronales Artificiales (RNA) se ha centrado en el desarrollo de nuevos algoritmos de aprendizaje, la exploración de nuevas arquitecturas de redes neuronales y la expansión de nuevos campos de aplicación. Sin embargo, se ha dedicado poca atención a desarrollar procedimientos que permitan comprender la naturaleza de las representaciones internas generadas por la red para responder ante una determinada tarea. En el presente trabajo, se plantea un doble objetivo: a) revisar los diferentes métodos interpretativos propuestos hasta el momento para determinar la importancia o efecto de cada variable de entrada sobre la salida de una red perceptrón multicapa, b) validar un nuevo método, denominado análisis de sensibilidad numérico (NSA, numeric sensitivity analysis), que permite superar las limitaciones de los métodos anteriormente propuestos. Los resultados obtenidos mediante simulación ponen de manifiesto que el método NSA es el procedimiento que, en términos globales, mejor describe el efecto de las variables de entrada sobre la salida de la red neuronal.
PALABRAS CLAVE: *Redes neuronales artificiales, análisis de sensibilidad.*

Abstract

ARTIFICIAL NEURAL NETWORKS: OPEN THE BLACK BOX. The biggest effort in the research about Artificial Neural Networks (ANN) has been centered in the development of new learning algorithms, the exploration of new architectures of neural networks and the expansion of new application fields. However, little attention has been dedicated on to develop procedures that allow to understand the nature of the internal representations generated by the neural network to respond to a certain task. The present work outlined two objectives: a) to revise the different interpretative methods proposed until the moment to determine the importance or effect of each input variable on the output of a multilayer perceptron, b) to validate a new method, called numeric sensitivity analysis (NSA), that allows to overcome the limitations of the previously proposed methods. The results obtained by simulation show that the NSA method is, in general terms, the best procedure for the analysis of the effect of the input variables on the neural network output.
KEY WORDS: *artificial neural networks, sensitivity analysis.*

En los últimos quince años, las redes neuronales artificiales (RNA) han emergido como una potente herramienta para el modelado estadístico orientada principalmente al reconocimiento de patrones -tanto en la vertiente de clasificación como de predicción. Las RNA poseen una serie de características admirables, tales como la habilidad para procesar datos con ruido o incompletos, la alta tolerancia a fallos que permite a la red operar satisfactoriamente con neuronas o conexiones dañadas y la capacidad de responder en tiempo real debido a su paralelismo inherente.

Actualmente, existen unos 40 paradigmas de RNA que son usados en diversos campos de aplicación (Sarle, 1998). Entre estos paradigmas, el más ampliamente utilizado es el perceptrón multicapa asociado al algoritmo de aprendizaje *backpropagation error* (propagación del error hacia atrás), también denominado *red backpropagation* (Rumelhart, Hinton y Williams, 1986). La popularidad del perceptrón

[1]Dirección postal: Juan José Montaño Moreno. Facultad de Psicología. Universidad de las Islas Baleares. Carretera de Valldemossa, Km. 7,5. 07071 Palma de Mallorca (Spain). *e-mail:* juanjo.montano@uib.es

multicapa se debe principalmente a que es capaz de actuar como un aproximador universal de funciones (Funahashi, 1989; Hornik, Stinchcombe y White, 1989). Más concretamente, una red conteniendo al menos una capa oculta con suficientes unidades no lineales puede aprender cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida. Esta propiedad convierte a las redes perceptrón multicapa en herramientas de propósito general, flexibles y no lineales; mostrando un rendimiento superior respecto a los modelos estadísticos clásicos en numerosos campos de aplicación.

El mayor esfuerzo en la investigación sobre RNA se ha centrado en el desarrollo de nuevos algoritmos de aprendizaje, la exploración de nuevas arquitecturas de redes neuronales y la expansión de nuevos campos de aplicación. Sin embargo, se ha dedicado poca atención a desarrollar procedimientos que permitan comprender la naturaleza de las representaciones internas generadas por la red para responder ante un problema determinado. En lugar de eso, las RNA se han presentado al usuario como una especie de "caja negra" cuyo complejísimo trabajo, de alguna forma mágico, transforma las entradas en salidas predichas. En otras palabras, no se puede saber inmediatamente cómo los pesos de la red o los valores de activación de las neuronas ocultas están relacionados con el conjunto de datos manejados. Así, a diferencia de los modelos estadísticos clásicos, no parece tan evidente conocer en una red el efecto que tiene cada variable explicativa sobre la/s variable/s de respuesta.

Esta percepción acerca de las RNA como una "caja negra", sin embargo, no es del todo cierta. De hecho, desde finales de los años 80 han surgido diversos intentos por desarrollar una metodología que permitiera interpretar lo aprendido por la red, aunque son escasos los trabajos orientados a la validación de tales procedimientos. Estos esfuerzos no tienen únicamente por objeto determinar las variables de entrada con mayor peso o importancia sobre la salida de la red, sino también identificar y eliminar del modelo las variables redundantes o irrelevantes, ésto es, aquellas variables que pueden expresarse en términos de otras variables de entrada o aquellas que simplemente no contribuyen en la predicción. Este último aspecto no es trivial en el campo de las RNA, debido a que las RNA –al igual que otros tipos de modelado–, se ven afectadas por la denominada maldición de la dimensionalidad (*curse of dimensionality*) (Bishop, 1995; Sarle, 1998). Esto significa que el número de datos necesarios para especificar una función, en general, crece exponencialmente con la dimensión del espacio de entrada. Por tanto, la reducción de la dimensión del espacio de entrada mediante la eliminación de variables redundantes o irrelevantes permite trabajar con un menor número de datos, acelera el proceso de convergencia de los pesos de la red y, en base a las demostraciones de Baum y Haussler (1989), podemos tener la expectativa de obtener un error de generalización más bajo (Rzempoluck, 1998).

En el presente trabajo nos planteamos dos objetivos: a) revisar los diferentes métodos interpretativos propuestos hasta el momento para determinar la importancia o efecto de cada variable de entrada sobre la salida de una red perceptrón multicapa, b) presentar y validar un nuevo método, denominado análisis de sensibilidad numérico (NSA, *numeric sensitivity analysis*), que permite superar algunas limitaciones de los métodos anteriormente propuestos.

El perceptrón multicapa

Consideremos un perceptrón multicapa compuesto por una capa de entrada, una capa oculta y una capa de salida como el mostrado en la figura 1. Un patrón de entrada p formado por un conjunto de valores en las variables de entrada x_i está representado por el vector $X_p : x_{p1}, \ldots, x_{pi}, \ldots, x_{pN}$. Por su parte, w_{ij} es el peso de conexión desde la neurona de entrada i a la neurona oculta j y v_{jk} es el peso de conexión desde la neurona oculta j a la neurona de salida k. Respecto a las señales de entrada y los valores de activación de las neuronas, net_{pj} y net_{pk} son las entradas netas que reciben las neuronas ocultas y de salida para un patrón p dado, respectivamente; b_{pj} e y_{pk} son los valores de activación o de salida de las neuronas ocultas y de salida, respectivamente, para un patrón p dado, como resultado de aplicar una función de activación $f(\cdot)$ sobre la entrada neta de la neurona.

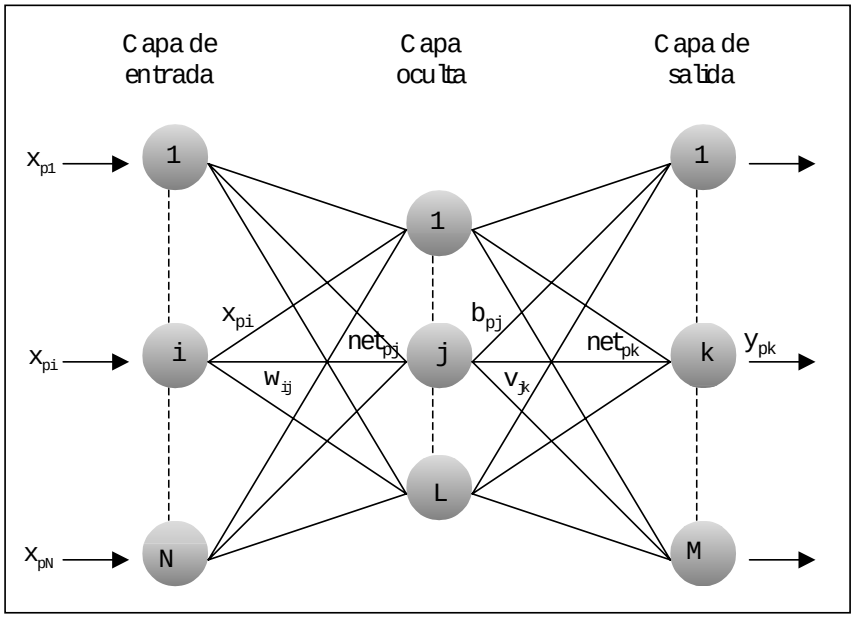


Figura 1. Perceptrón multicapa compuesto por una capa de entrada, una capa oculta y una capa de salida.

Abriendo la caja negra

En el presente apartado se exponen dos tipos de metodologías generales que permiten conocer lo que ha aprendido un perceptrón multicapa con una capa oculta a partir del valor de los pesos y los valores de activación de las neuronas, esto es, lo que se pretende es conocer el efecto o importancia de cada variable de entrada sobre la salida de la red. Estas dos metodologías corresponden al análisis basado en la magnitud de los pesos y al análisis de sensibilidad.

Análisis basado en la magnitud de los pesos

El análisis basado en la magnitud de los pesos agrupa aquellos procedimientos que se basan exclusivamente en los valores almacenados en la matriz estática de pesos con el propósito de determinar la influencia relativa de cada variable de entrada sobre cada una de las salidas de la red. Este tipo de análisis tiene su origen en el examen de los pesos de conexión entre las neuronas de entrada y ocultas. Así, se podría decir que las entradas con pesos de valor absoluto alto son importantes, mientras que aquellas cuyo valor de pesos es próximo a cero no son importantes. Sin embargo, este método es poco fiable para medir la importancia de las variables de entrada, debido a que la presencia de pesos altos en las conexiones entre la capa de entrada y oculta, no necesariamente significa que la entrada sea importante, y viceversa (Masters, 1993). Este método inicial ha dejado paso a expresiones matemáticas más elaboradas que tienen en cuenta no solo los pesos de conexión entre la capa de entrada y oculta, sino también los pesos de conexión entre la capa oculta y de salida.

En este sentido, se han propuesto diferentes ecuaciones basadas en la magnitud de los pesos (Yoon, Brobst, Bergstresser y Peterson, 1989; Baba, Enbutu y Yoda, 1990; Garson, 1991a; Garson, 1991b; Yoon, Swales y Margavio, 1993; Milne, 1995; Gedeon, 1997; Tsaih, 1999), aunque todas ellas se caracterizan por calcular el producto de los pesos w_{ij} y v_{jk} para cada una de las neuronas ocultas y obtener el sumatorio de los productos calculados. A continuación, se presenta una de las ecuaciones más utilizadas, la propuesta por Garson (Garson, 1991a, 1991b; Modai, Saban, Stoler, Valevski y Saban, 1995):

$$Q_{ik} = \frac{\sum_{j=1}^{L} \left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk} \right)}{\sum_{i=1}^{N} \left(\sum_{j=1}^{L} \left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk} \right) \right)} \tag{1}$$

donde $\sum_{r=1}^{N} w_{rj}$ es la suma de los pesos de conexión entre las i neuronas de entrada y la neurona oculta j.

En esta ecuación debemos tener en cuenta, por una parte, que el valor de los pesos se usa en valor absoluto para que los pesos positivos y negativos no se cancelen y, por otra parte, que el valor del umbral de las neuronas ocultas y de salida no se tienen en cuenta, asumiendo que su inclusión no afecta al resultado final (Garson, 1991a). El índice Q_{ik} representa el porcentaje de influencia de la variable de entrada i sobre la salida k, en relación a las demás variables de entrada, de forma que la suma de este índice para todas las variables de entrada debe dar como valor el 100%.

Análisis de sensibilidad

El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida y_k o en el error cometido debido al cambio que se produce en una entrada x_i. Así, cuanto mayor efecto se observe sobre la salida, mayor sensibilidad podemos deducir que presenta respecto a la entrada. A continuación, se presentan diferentes formas de realizar el análisis de sensibilidad.

Análisis de sensibilidad basado en el error

Una forma de aplicar el análisis de sensibilidad consiste en analizar el efecto producido sobre el error debido a cambios en la entrada. Normalmente, la función de error que se utiliza es la raíz cuadrada de la media cuadrática del error (RMC error) que viene dada por la siguiente expresión:

$$RMC_{error} = \sqrt{\frac{\sum_{p=1}^{P} \sum_{k=1}^{M} (d_{pk} - y_{pk})^2}{P \cdot M}} \tag{2}$$

donde d_{pk} es la salida deseada para el patrón p en la neurona de salida k.

La aplicación de este tipo de análisis sobre un conjunto de datos –en general, el grupo de entrenamiento–, consiste en ir variando el valor de una de las variables de entrada a lo largo de todo su rango mediante la aplicación de pequeños incrementos, mientras se mantienen los valores originales de las demás variables de entrada (Frost y Karri, 1999). Una manera sencilla de determinar el tamaño de los incrementos en las variables de entrada se basa en dividir el rango de la variable por el número de patrones con el que se cuenta. Una vez aplicados los incrementos a una determinada variable de entrada, se procede a entrenar la red neuronal calculando el valor de RMC error. Siguiendo este procedimiento para todas las variables de entrada, se puede establecer una ordenación en cuanto a importancia sobre la salida. Así, la variable de entrada que proporcione el mayor RMC error será la variable con más influencia en la variable de respuesta, mientras que la variable de entrada con menor RMC error asociado será la que menos contribuya en la predicción de la red.

Existen otras variantes de este método como, por ejemplo, evaluar el error cometido por la red restringiendo la entrada de interés a un valor fijo (por ejemplo, el valor promedio) para todos los patrones o directamente eliminando esa entrada (Masters, 1993). Si el error aumenta sensiblemente ante el cambio provocado, se puede concluir que la entrada es importante.

Análisis de sensibilidad basado en la salida

Una forma más común de realizar el análisis de sensibilidad consiste en estudiar el efecto que se observa directamente en una variable de salida debido al cambio que se produce en una variable de entrada. Esta aproximación ha sido aplicada de forma intuitiva en la identificación de variables relevantes en un variado número de tareas como la predicción del comportamiento de la bolsa (Bilge, Refenes, Diamond y Shadbolt, 1993), la predicción de las auto-expectativas negativas en niños (Reid, Nair, Kashani y Rao, 1994; Kashani, Nair, Rao, Nair y Reid 1996), el análisis de supervivencia (De Laurentiis y Ravdin, 1994), la predicción del resultado de un tratamiento psiquiátrico (Modai, Saban, Stoler, Valevski y Saban, 1995) y la predicción de resultados farmacológicos (Opara, Primozic y Cvelbar, 1999). De esta forma, sobre la red entrenada se fija el valor de todas las variables de entrada a su valor medio y, a continuación, se puede optar por añadir ruido o ir variando el valor de una de las variables a lo largo de todo su rango mediante la aplicación de pequeños incrementos. Esto permite registrar los cambios producidos en la salida de la red y aplicar sobre estos cambios un índice resumen que dé cuenta de la magnitud del efecto de las variaciones producidas en la entrada x_i sobre la salida y_k.

Esta forma de proceder es sencilla de aplicar, sin embargo, desde esta aproximación se debe tomar una decisión bastante arbitraria acerca de la cantidad de ruido a añadir o de la magnitud del incremento y también del valor al que quedan fijadas las demás variables. A continuación, se presentan dos métodos basados en el análisis de sensibilidad sobre la salida que gozan de un fundamento matemático más sólido: la matriz de sensibilidad Jacobiana y el método de sensibilidad numérico.

Matriz de sensibilidad Jacobiana

El algoritmo de aprendizaje *backpropagation* (Rumelhart, Hinton y Williams, 1986) aplicado a un perceptrón multicapa se basa en el cálculo de la derivada parcial del error con respecto a los pesos para averiguar qué dirección tomar para modificar los pesos con el fin de reducir el error de forma iterativa. Mediante un procedimiento similar, los elementos que componen la matriz Jacobiana S proporcionan, de forma analítica, una medida de la sensibilidad de las salidas a cambios que se producen en cada una de las variables de entrada. En la matriz Jacobiana S -de orden IxK–, cada fila representa una entrada de la red y cada columna representa una salida de la red, de forma que el elemento S_{ik} de la matriz representa la sensibilidad de la salida k con respecto a la entrada i. Cada uno de los elementos S_{ik} se obtiene calculando la derivada parcial de una salida y_k con respecto a una entrada x_i, esto es, $\frac{\partial y_k}{\partial x_i}$. En este caso, la derivada parcial representa la pendiente instantánea de la función subyacente entre x_i e y_k para unos valores dados en ambas variables. Aplicando la regla de la cadena sobre $\frac{\partial y_k}{\partial x_i}$ tenemos que:

$$S_{ik} = \frac{\partial y_k}{\partial x_i} = f'(net_k) \sum_{j=1}^{L} v_{jk} f'(net_j) w_{ij} \tag{3}$$

Si las neuronas de la capa oculta y de salida utilizan la función de activación sigmoidal logística, la expresión final de cálculo de la sensibilidad de la salida k con respecto a la entrada i sería:

$$S_{ik} = y_k(1 - y_k) \sum_{j=1}^{L} v_{jk} b_j(1 - b_j) w_{ij} \tag{4}$$

Así, cuanto mayor sea el valor absoluto de S_{ik}, más importante es x_i en relación a y_k. El signo de S_{ik} indica si el cambio observado en y_k va en la misma dirección o no que el cambio provocado en x_i. Cuando la discrepancia entre la salida y_k calculada por la red y la salida deseada d_k para un patrón dado es mínima, el término derivativo $f'(net_k)$ es próximo a cero con funciones sigmoidales y, como consecuencia, el valor de la derivada parcial queda anulado. Con el fin de solucionar este problema, algunos autores (Lisboa, Mehridehnavi y Martin, 1994; Rzempoluck, 1998) proponen suprimir el término derivativo $f'(net_k)$ asumiendo que no afecta a la comparación entre las sensibilidades de las diferentes entradas respecto a la salida. Una solución alternativa que proponemos podría consistir en utilizar la función lineal en las neuronas de salida. De esta forma, evitamos la cancelación del valor de la derivada parcial ya que la derivada de la función lineal es igual a la unidad.

Como se puede observar, los valores de la matriz Jacobiana dependen no solo de la información aprendida por la red neuronal, que está almacenada de forma dis-

tribuida y estática en las conexiones w_{ij} y v_{jk}, sino también de la activación de las neuronas de la capa oculta y de salida que, a su vez, dependen de las entradas de la red. Como diferentes patrones de entrada pueden proporcionar diferentes valores de pendiente, la sensibilidad necesita ser evaluada a partir de todo el conjunto de entrenamiento. Considerando el valor de sensibilidad entre i y k para el patrón X_p como $S_{ik}(p)$, podemos definir la sensibilidad a partir de la esperanza matemática, $E(S_{ik}(p))$, y la desviación estándar, $SD(S_{ik}(p))$, que pueden ser calculadas mediante:

$$E(S_{ik}(p)) = \frac{\sum_{p=1}^{P} S_{ik}(p)}{P} \tag{5}$$

$$SD(S_{ik}(p)) = \sqrt{\frac{\sum_{p=1}^{P} \left(S_{ik}(p) - E(S_{ik}(p))\right)^2}{P-1}} \tag{6}$$

El cálculo de la matriz Jacobiana ha sido expuesto en multitud de ocasiones (Hwang, Choi, Oh y Marks, 1991; Hashem, 1992; Fu y Chen, 1993; Zurada, Malinowski y Cloete, 1994; Bishop, 1995; Rzempoluck, 1998) y ha sido aplicado en campos tan variados como el reconocimiento de imágenes (Takenaga, Abe, Takatoo, Kayama, Kitamura y Okuyama, 1991), la ingeniería (Guo y Uhrig, 1992; Bahbah y Girgis, 1999), la meteorología (Castellanos, Pazos, Ríos y Zafra, 1994) y la medicina (Harrison, Marshall y Kennedy, 1991; Engelbrecht, Cloete y Zurada, 1995; Rambhia, Glenny y Hwang, 1999).

Respecto a los estudios de validación de este método, Gedeon (1997) realizó recientemente una comparación entre el análisis basado en la magnitud de los pesos y el análisis de sensibilidad basado en el cálculo de la matriz Jacobiana, mostrando este último tipo de análisis un mayor acuerdo con la técnica de ir eliminando cada vez una variable de entrada y observar el efecto que tiene en el rendimiento de la red neuronal. Estos resultados parecen indicar que las técnicas basadas en propiedades dinámicas como el análisis de sensibilidad son más fiables que las técnicas basadas en propiedades estáticas.

Método de sensibilidad numérico

En los apartados anteriores, se han descrito los diferentes métodos propuestos para analizar la importancia que tienen las variables explicativas sobre la/s variable/s de respuesta en una red perceptrón multicapa. Tales métodos han demostrado su utilidad en determinadas tareas de predicción, sin embargo, cuentan con una serie de limitaciones que pasamos a comentar.

El análisis basado en la magnitud de los pesos no ha demostrado ser sensible a la hora de ordenar las variables de entrada en función de su importancia sobre la salida (Garson, 1991a; Sarle, 2000) y, en los estudios comparativos (Gedeon, 1997), el análisis de sensibilidad basado en el cálculo de la matriz Jacobiana ha demostrado ser siempre superior. Por su parte, el análisis de sensibilidad consistente en añadir

incrementos o perturbaciones se basa en la utilización de variables de entrada cuya naturaleza es continua, ya que no sería del todo correcto añadir incrementos a variables nominales, esto es, variables que toman valores discretos (Hunter, Kennedy, Henry y Ferguson, 2000). Por su parte, la versión analítica del análisis de sensibilidad, el cálculo de la matriz Jacobiana, parte del supuesto de que todas las variables implicadas en el modelo son continuas (Sarle, 2000). Este supuesto limita el número de campos de aplicación de las RNA en las Ciencias del Comportamiento y de la Salud donde es muy común el manejo de variables discretas (por ejemplo, género: 0 = varón, 1 = mujer ó estatus: 0 = sano, 1 = enfermo).

En este trabajo, se presenta un nuevo método, denominado análisis de sensibilidad numérico (NSA, *numeric sensitivity analysis*), que permite superar las limitaciones comentadas de los métodos anteriores. Este nuevo método se basa en el cálculo de las pendientes que se forman entre entradas y salidas, sin realizar ningún supuesto acerca de la naturaleza de las variables y respetando la estructura original de los datos.

Para analizar el efecto de una variable de entrada x_i sobre una variable de salida y_k mediante el método NSA, en primer lugar, debemos ordenar ascendentemente los p patrones a partir de los valores de la variable de entrada x_i. En función de tal ordenación, se genera un número G determinado de grupos de igual o aproximado tamaño. El número idóneo de grupos dependerá del número de patrones disponible y de la complejidad de la función que se establece entre la entrada y la salida, aunque en la mayoría de casos será suficiente un valor de $G = 30$ o similar. Para cada grupo formado se calcula la media aritmética de la variable x_i y la media aritmética de la variable y_k. A continuación, se obtiene el índice NSA basado en el cálculo numérico de la pendiente formada entre cada par de grupos consecutivos, g_r y g_{r+1}, de x_i sobre y_k mediante la siguiente expresión:

$$NSA_{ik}(g_r) \equiv \frac{\bar{y}_k(g_{r+1}) - \bar{y}_k(g_r)}{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)} \tag{7}$$

donde

- $\bar{x}_i(g_r)$ y $\bar{x}_i(g_{r+1})$ son las medias de la variable x_i correspondientes a los grupos g_r y g_{r+1}, respectivamente,
- $\bar{y}_k(g_r)$ e $\bar{y}_k(g_{r+1})$ son las medias de la variable y_k correspondientes a los grupos g_r y g_{r+1}, respectivamente.

Una vez calculados los $G-1$ valores NSA, se puede obtener el valor de la esperanza matemática del índice NSA o pendiente entre la variable de entrada i y la variable de salida k mediante:

$$E(NSA_{ik}(g_r)) = \sum_{r=1}^{G-1} NSA_{ik}(g_r) \cdot f(NSA_{ik}(g_r)) = \frac{\bar{y}_k(g_G) - \bar{y}_k(g_1)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)} \tag{8}$$

donde

- $f(NSA_{ik}(g_r)) \equiv \frac{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)}$ representa la función de probabilidad del índice NSA,

- $\bar{x}_i(g_G)$ y $\bar{x}_i(g_1)$ son los valores promedio de la variable x_i para el último grupo g_G y el primer grupo g_1, respectivamente,
- $\bar{y}_k(g_G)$ e $\bar{y}_k(g_1)$ son los valores promedio de la variable y_k para el grupo g_G y el grupo g_1, respectivamente.

Cuando la variable de entrada x_i es binaria, la esperanza del índice NSA se obtiene calculando la media de la variable y_k cuando la variable x_i toma el valor mínimo y la media de la variable y_k cuando la variable x_i toma el valor máximo, y aplicando la siguiente expresión:

$$E(NSA_{ik}(g_r)) = \frac{\bar{y}_k(x_{i\max}) - \bar{y}_k(x_{i\min})}{x_{i\max} - x_{i\min}} \tag{9}$$

donde

- $x_{i\max}$ y $x_{i\min}$ son los valores máximo y mínimo que toma la variable x_i, respectivamente,
- $\bar{y}_k(x_{i\max})$ e $\bar{y}_k(x_{i\min})$ son la media de la variable y_k cuando la variable x_i toma el valor máximo y la media de la variable y_k cuando la variable x_i toma el valor mínimo, respectivamente.

El valor de la esperanza matemática del índice NSA representa el efecto promedio que tiene un incremento de x_i sobre y_i. Cuando la variable de entrada es binaria, la esperanza matemática representa el efecto promedio provocado por el cambio del valor mínimo al valor máximo en la variable x_i. Al igual que en el caso de la matriz Jacobiana, cuanto mayor sea el valor absoluto de $E(NSA_{ik}(g_r))$, más importante es x_i en relación a y_k. El signo de $E(NSA_{ik}(g_r))$ indica si el cambio observado en y_k va en la misma dirección o no que el cambio provocado en x_i.

Los estudios piloto en los que hemos aplicado el método NSA muestran que la forma más adecuada de representar las variables de entrada de naturaleza discreta (binarias o politómicas) es mediante la utilización de codificación ficticia, tal como se hace habitualmente en el modelado estadístico. La introducción de los valores originales de una variable politómica (por ejemplo, nivel social: bajo = 1, medio = 2 y alto = 3) en la red neuronal, no permite reflejar de forma adecuada el efecto o pendiente que tiene el cambio de una categoría a otra categoría. Por otro lado, es conveniente que las variables de entrada y salida sean reescaladas al mismo rango de posibles valores (por ejemplo, entre 0 y 1). Esto último evita posibles sesgos en la obtención del índice NSA debido a la utilización de escalas de medida diferentes entre las variables de entrada y, además, permite obtener un valor de esperanza matemática estandarizado a diferencia de la matriz Jacobiana. Así, el rango de posibles valores que puede adoptar $E(NSA_{ik}(g_r))$ oscila entre -1 y $+1$. Estos dos límites indicarían un efecto máximo de la variable de entrada sobre la salida, con una relación negativa en el primer caso (-1) y con una relación positiva en el segundo caso ($+1$). Los valores iguales o próximos a cero indicarían ausencia de efecto de la variable de entrada.

El cálculo de la desviación estándar del índice NSA, cuando las variables implicadas son de naturaleza continua, se puede realizar mediante:

$$SD(NSA_{ik}(g_r)) = \sqrt{E(NSA_{ik}^2(g_r)) - (E(NSA_{ik}(g_r)))^2} \tag{10}$$

El valor de la desviación estándar se debe interpretar como el grado de oscilaciones que ha sufrido la pendiente que se establece entre x_i e y_k, de manera que a mayor valor de la desviación estándar, mayor comportamiento caótico o aleatorio tiene la función entre las dos variables implicadas.

Por último, es conveniente aportar junto a los índices esperanza matemática y desviación estándar, la representación gráfica de la pendiente entre cada par de variables de entrada y salida. Esto es especialmente útil en aquellos casos en los que el valor de la esperanza matemática puede enmascarar la función subyacente entre la variable de entrada y la variable de salida.

Una vez descritos los principales métodos interpretativos dirigidos al análisis del efecto de las variables de entrada en una red perceptrón multicapa, a continuación, nos proponemos realizar un estudio comparativo acerca de su rendimiento. Los métodos bajo estudio son: análisis basado en la magnitud de los pesos a través del método de Garson (1991a), análisis de sensibilidad basado en el cálculo del incremento observado en la función RMC error al eliminar una variable de entrada, análisis de sensibilidad basado en el cálculo de la matriz Jacobiana y, finalmente, el método NSA.

Método

Materiales y procedimiento

Se han generado mediante simulación cuatro matrices de datos compuesta cada una de ellas por 1000 registros y cuatro variables con rango entre 0 y 1. Las tres primeras variables (X1, X2 y X3) actuan como variables predictoras o variables de entrada a la red, mientras que la última variable (Y) es una función de las variables predictoras y actua como variable de salida. El valor del coeficiente de correlación de Pearson entre las variables predictoras oscila entre 0 y 0.40. En todos los casos, la variable X1 no tiene ningún tipo de contribución o efecto en la salida Y de la red, seguida de la variable X2 con un efecto intermedio y la variable X3 que presenta el mayor efecto sobre la salida de la red. A fin de analizar el comportamiento de los diferentes métodos dependiendo del tipo de variable implicada, se ha manipulado en cada una de las matrices la naturaleza de las variables de entrada. A continuación, se presenta una descripción detallada de cada matriz utilizada:

Matriz 1: Variables cuantitativas. Las variables de entrada son de naturaleza cuantitativa con distribución normal. Para generar la variable Y como función de las variables de entrada se utilizó la siguiente expresión:

$$Y = \tanh(X2) - \exp(2.4 \cdot X3) + Error\ N(0, 0.1) \tag{11}$$

De esta forma, X1 no contribuye en la variable Y. La función que se establece entre X2 y la variable de salida es la tangente hiperbólica con rango entre -1 y 1 para la variable de salida, mientras que la función entre X3 y la variable de salida es de tipo exponencial negativo con rango entre -1 y -10 para la variable de salida. Por último, se añadió un error aleatorio con distribución normal, media 0 y desviación estándar 0.1. La variable de salida resultante se reescaló a valores entre 0 y 1.

Matriz 2: Variables discretas binarias. Las variables de entrada y salida son de naturaleza discreta binaria con valores 0 y 1. La relación entre las entradas y la salida se determinó a partir del coeficiente Phi. Así, los valores del coeficiente Phi entre X1, X2, X3 y la salida fueron -0.001, 0.346 y 0.528, respectivamente.

Matriz 3: Variables discretas politómicas. Las variables de entrada son de naturaleza discreta politómica con valores 1, 2 y 3, mientras que la variable de salida es discreta binaria con valores 0 y 1. La relación entre las entradas y la salida se determinó a partir del coeficiente V de Cramer. Así, los valores del coeficiente V entre X1, X2, X3 y la salida fueron 0.023, 0.380 y 0.593, respectivamente. Cada variable predictora se codificó mediante la utilización de dos variables ficticias binarias (valor 1 = 0 0, valor 2 = 1 0 y valor 3 = 0 1), de forma que cada variable estaba representada por dos neuronas de entrada.

Matriz 4: Variables cuantitativas y discretas. La variable X1 es de naturaleza cuantitativa con distribución normal, la variable X2 es discreta binaria con valores 0, 1 y la variable X3 es continua con distribución normal. Para generar la variable Y como función de las variables de entrada se utilizó la siguiente expresión:

$$\begin{cases} \text{Si } X2 = 0, \text{ entonces} & Y = 1 + 1.1 \times \exp(X3) + Error\ N(0, 0.1) \\ \text{Si } X2 = 1, \text{ entonces} & Y = 1.1 \times \exp(X3) + Error\ N(0, 0.1) \end{cases} \tag{12}$$

De esta forma, X1 no contribuye en la variable Y. Por su parte, cuando X2 toma el valor 0, la variable Y se incrementa en una unidad. La función que se establece entre X3 y la variable Y es de tipo exponencial con rango entre 1 y 3 para la variable de salida. Al igual que en el caso de la matriz 1, se añadió un error aleatorio con distribución normal, media 0 y desviación estándar 0.1. La variable de salida resultante se reescaló a valores entre 0 y 1.

Para llevar a cabo el estudio, se utilizó el programa informático *Sensitivity Neural Network 1.0*, creado por nosotros, que, a través de un interfaz de sencillo manejo, permite simular el comportamiento de un perceptrón multicapa entrenado con el algoritmo de aprendizaje *backpropagation* e incorpora como novedad los métodos interpretativos descritos.

Para el entrenamiento de las redes neuronales, cada matriz de datos fue dividida en tres grupos: 500 patrones actuaron como conjunto de entrenamiento, 250 patrones actuaron como conjunto de validación y 250 patrones actuaron como conjunto de test. Las redes neuronales simuladas estaban compuestas por tres o seis neuronas de entrada dependiendo de la matriz de datos, dos neuronas ocultas y una neurona de salida. Como funciones de activación se utilizó la función tangente hiperbólica en las neuronas ocultas y la función lineal en la neurona de salida. Como parámetros de aprendizaje se utilizó un valor de 0.25 para la tasa de aprendizaje y un valor de 0.8 para el factor momentum. Los pesos de conexión y umbral fueron inicializados con diferentes valores semilla. La red neuronal que obtuvo el mejor rendimiento ante el conjunto de validación de la matriz correspondiente, fue la seleccionada para pasar a la fase de test. Los cuatro modelos de red finalmente obtenidos, mostraron un buen ajuste ante los datos de test.

Resultados

La Tabla 1 muestra los resultados obtenidos tras aplicar los métodos interpretativos sobre los modelos de red seleccionados en la fase de validación.

Tabla 1. Resultados obtenidos tras aplicar los métodos interpretativos sobre los modelos de red

Métodos interpretativos						
	Método de Garson	Δ error	Matriz de sensibilidad Jacobiana		Análisis de sensibilidad numérico	
			Media	S.D.	Media	S.D.
Matriz 1						
X1	1.102	0.001	0.001	0.003	0.004	1.934
X2	38.980	0.043	0.388	0.027	0.468	1.560
X3	59.920	0.113	-0.786	0.311	-0.865	0.923
Matriz 2						
X1	24.890	0.001	0.007	0.028	-0.001	–
X2	30.140	0.037	0.339	0.048	0.345	–
X3	44.960	0.081	0.501	0.069	0.527	–
Matriz 3						
X1						
X1V1	5.422	–	-0.022	0.032	-0.021	–
X1V2	6.650	–	0.023	0.054	0.002	–
Suma	12.072	0.001	0.045		0.023	
X2						
X2V1	13.630	–	0.306	0.229	0.080	–
X2V2	16.560	–	0.450	0.317	0.303	–
Suma	30.190	0.050	0.756		0.383	
X3						
X3V1	7.049	–	0.302	0.186	0.027	–
X3V2	50.690	–	0.464	0.534	0.531	–
Suma	57.739	0.109	0.766		0.558	
Matriz 4						
X1	17.020	0.000	0.001	0.014	0.094	2.810
X2	29.930	0.132	-0.312	0.006	-0.325	–
X3	53.050	0.076	0.599	0.086	0.585	1.770

Se puede observar que, en general, los cuatro métodos analizados establecen correctamente la jerarquía entre las variables de entrada en función de su importancia.

Un análisis en detalle permite observar que los métodos se comportan de forma correcta cuando las variables de entrada son cuantitativas (matriz 1), y cuando son cuantitativas y discretas (matriz 4), a excepción del método basado en el análisis del incremento del error el cual otorga más importancia a X2 frente a X3 en la matriz 4. Por tanto, este método no establece de forma correcta la jerarquía de importancia entre las variables de entrada. Por su parte, el método de Garson sobrevalora el efecto de la variable X1 en la matriz 4 proporcionando para esta variable un 17.02% de importancia relativa, cuando en realidad su efecto es nulo. Finalmente, el cálculo

de la matriz Jacobiana y el método NSA proporcionan valores promedio muy similares con las matrices 1 y 4. También se puede observar que estos dos métodos identifican aquellos casos en los que la relación entre entrada y salida es negativa.

Cuando las variables de entrada son discretas binarias (matriz 2), se puede observar que el método NSA es el que representa con más exactitud la realidad, debido a que este índice coincide con bastante precisión con los valores reales del índice Phi entre las variables de entrada y la salida de la red. Por otra parte, se puede observar que el cálculo de la matriz Jacobiana, proporciona valores promedio próximos al índice Phi, pero no con la exactitud dada por el método NSA. Este aspecto se puede apreciar mejor observando la tabla 2 donde se proporcionan los valores Phi y los promedios de la matriz Jacobiana y el método NSA para la matriz 2. Por su parte, tal como expresa la tabla 1, el método de Garson en este caso también sobrevalora la importancia de X1, proporcionando un valor de 24.89%.

Tabla 2. Valores Phi, V e índices obtenidos mediante la matriz de sensibilidad Jacobiana y el análisis de sensibilidad numérico para las matrices 2 y 3.

	X1	X2	X3
Matriz 2			
Phi	-0.001	0.346	0.528
Jacobiana	0.007	0.339	0.501
Numérico	-0.001	0.345	0.527
Matriz 3			
V	0.023	0.380	0.593
Jacobiana	0.045	0.756	0.766
Numérico	0.023	0.383	0.558

Cuando las variables de entrada son discretas politómicas (matriz 3), se puede comprobar que el método NSA también es el que mejor se ajusta a la realidad, debido a que es el método que más se aproxima a los valores del índice V. Para ello, como se puede observar en la Tabla 1, se calcula la suma de los valores absolutos obtenidos con las dos variables ficticias que representan cada variable predictora. La Tabla 2 permite comparar el valor real del índice V con los valores promedio de la matriz Jacobiana y el método NSA para la matriz 3. En este caso, la matriz Jacobiana otorga prácticamente la misma importancia a X2 y a X3 con valores 0.756 y 0.766, respectivamente, cuando en realidad X3 es más importante que X2. Al igual que en los casos anteriores, el método de Garson sobrevalora la importancia de X1, proporcionando un valor de 12.07%.

Respecto al grado de fiabilidad, se puede observar que el cálculo de la matriz Jacobiana proporciona en general una gran precisión en la estimación de la media en función del valor de la desviación estándar. Recordemos que en el caso del método NSA, el valor de la desviación estándar se debe interpretar como el grado de oscilación que ha sufrido la pendiente que se establece entre una entrada cuantitativa y la salida. Así, se puede observar en la matriz 1 (ver Tabla 1) que cuanto mayor aleatoriedad existe en la función que se establece entre las dos variables implicadas, mayor es el valor de la desviación estándar de la pendiente.

Por otra parte, con el método NSA la interpretación del efecto de una variable es mucho más sencilla ya que la media que proporciona está acotada a valores entre -1 y 1. Recordemos que esta condición se cumplirá, siempre que se acoten las variables

a un mismo rango de valores. En cambio, los valores promedio proporcionados por la matriz Jacobiana pueden oscilar, a nivel teórico, entre $-\infty$ y $+\infty$.

Finalmente, a modo de ejemplo, se presenta en la Figura 2 la representación gráfica del método NSA proporcionada por el programa *Sensitivity Neural Network 1.0* para la variable X3 de la matriz 1. Recordemos que en la simulación se determinó que la función subyacente entre esta variable de entrada y la salida era una función exponencial negativa. Se puede observar que la red neuronal ha aprendido correctamente la función existente entre ambas variables.

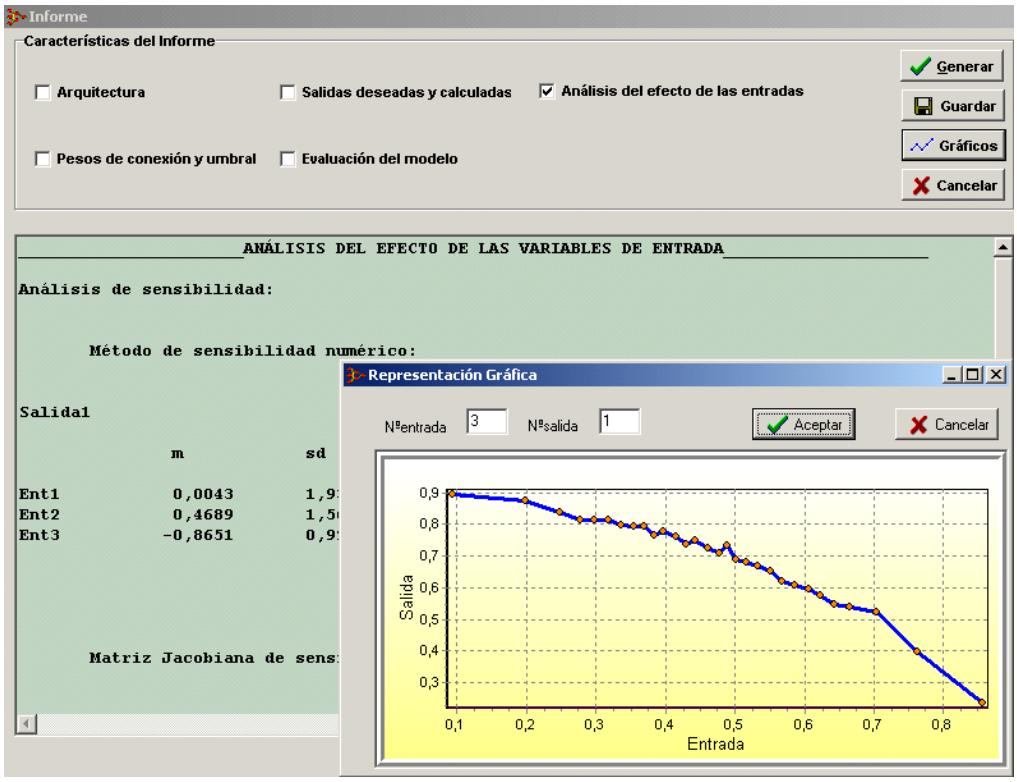


Figura 2. Representación gráfica del método NSA proporcionada por el programa Sensitivity Neural Network 1.0 para la variable X3 de la matriz 1.

Conclusiones

Las redes neuronales del tipo perceptrón multicapa han obtenido, en los últimos años, excelentes resultados como predictores en tareas de clasificación de patrones y estimación de variables cuantitativas. Sin embargo, este tipo de arquitectura no permite analizar, al menos directamente, el papel desempeñado por cada variable predictora sobre la salida de la red.

Se ha realizado un estudio comparativo acerca de la utilidad de cuatro métodos dirigidos a evaluar la importancia relativa de las variables de entrada en un perceptrón multicapa. Para ello, se han obtenido mediante simulación cuatro matrices de datos en donde se ha manipulado el grado de relación o importancia entre las variables y la naturaleza de las mismas.

De los resultados obtenidos se desprende una serie de ventajas en la aplicación del método NSA presentado en este trabajo, con respecto a los métodos anteriormente propuestos.

En primer lugar, el método NSA es el que, en cuanto a grado de generalidad, permite describir mejor el efecto o importancia de las variables de entrada sobre la salida. Se ha podido observar que el método NSA y la matriz Jacobiana proporcionan valores promedio similares cuando las variables implicadas son cuantitativas. La matriz Jacobiana es perfectamente válida en estos casos, teniendo en cuenta además la baja variabilidad que proporciona. En este sentido, trabajos anteriores corroboran esta conclusión (Harrison, Marshall y Kennedy, 1991; Takenaga, Abe, Takatoo, Kayama, Kitamura y Okuyama, 1991; Guo y Uhrig, 1992; Castellanos, Pazos, Ríos y Zafra, 1994; Engelbrecht, Cloete y Zurada, 1995; Bahbah y Girgis, 1999; Rambhia, Glenny y Hwang, 1999).

Sin embargo, cuando las variables implicadas son discretas (binarias y politómicas), el método NSA es el más adecuado debido a que los valores que proporciona se aproximan considerablemente al índice de asociación Phi en el caso de variables binarias y al índice de asociación V en el caso de variables politómicas.

El método de Garson sobrevalora en la mayoría de casos la importancia de variables que son irrelevantes para la salida de la red. La baja validez del método de Garson observada en nuestro estudio es coincidente con estudios anteriores (Gedeon, 1997; Sarle, 2000). Por su parte, el método basado en el cálculo del incremento observado en la función RMC error al eliminar una variable de entrada no ha sido capaz de establecer correctamente la jerarquía de importancia cuando las variables implicadas son cuantitativas y discretas (matriz 4). Además, este método no permite una interpretación sencilla respecto al grado de relación entre una variable de entrada y la salida de la red.

En segundo lugar, con el método NSA la interpretación del efecto de una variable es mucho más sencilla ya que el índice NSA que proporciona está acotado en el intervalo $[-1, 1]$ a diferencia de los valores que teóricamente puede proporcionar la matriz Jacobiana.

Por último, el método NSA incorpora un procedimiento que permite representar gráficamente la función aprendida por la red entre una variable de entrada y la salida. Esta representación gráfica aporta información relevante que complementa la información proporcionada por los índices numéricos, debido a que en muchos casos un índice de resumen no es suficiente para reflejar la función subyacente entre variables.

A modo de resumen, se puede decir que el método NSA es el que, de forma global, permite evaluar con mayor exactitud la importancia o efecto de las variables de entrada con independencia de su naturaleza (cuantitativa o discreta), superando las limitaciones de los métodos propuestos hasta el momento. Con este método se amplia considerablemente el número de campos de aplicación potenciales donde la utilización de variables discretas es muy común: medicina, sociología, psicología, biología, economía, etc. Ahora no solo podremos construir potentes instrumentos de predicción mediante redes perceptrón multicapa sino que a partir de las redes obtenidas también podremos analizar el impacto de las variables predictoras en cualquier conjunto de datos.

El método NSA junto a los métodos anteriormente propuestos se encuentran im-

plementados, como se ha comentado, en el programa *Sensitivity Neural Network 1.0*, creado por nuestro equipo. De esta forma, el usuario puede analizar las cualidades de cada uno de los métodos interpretativos con respecto a los demás para diferentes configuraciones de datos. Para este fin, *Sensitivity Neural Network 1.0* se encuentra disponible poniéndose en contacto con los autores via correo electrónico (`juanjo.montano@uib.es`).

Referencias

Baba, K., Enbutu, I. y Yoda, M. (1990). Explicit representation of knowledge acquired from plant historical data using neural network. En IEEE (Ed.), *Proceedings of the International Joint Conference on Neural Networks* (pp. 155-160). New York: IEEE.

Bahbah, A.G. y Girgis, A.A. (1999). Input feature selection for real-time transient stability assessment for artificial neural network (ANN) using ANN sensitivity analysis. En IEEE (Ed.), *Proceedings of the 21st International Conference on Power Industry Computer Applications* (pp. 295-300). Piscataway, NJ: IEEE.

Baum, E.B. y Haussler, D. (1989). What size net gives valid generalization? *Neural Computation, 1,* 151-160.

Bilge, U., Refenes, A.N., Diamond, C. y Shadbolt, J. (1993). Application of sensitivity analysis techniques to neural network bond forecasting. En A.N. Refenes (Ed.), *Proceedings of 1st International Workshop on Neural Networks in the Capital Markets* (p. 12). London: London Business School.

Bishop, C.M. (1995). *Neural networks for pattern recognition.* Oxford: Oxford University Press.

Castellanos, J., Pazos, A., Ríos, J. y Zafra, J.L. (1994). Sensitivity analysis on neural networks for meteorological variable forecasting. En J. Vlontzos, J.N. Hwang y E. Wilson (Eds.), *Proceedings of IEEE Workshop on Neural Networks for Signal Processing* (pp. 587-595). New York: IEEE.

De Laurentiis, M. y Ravdin, P.M. (1994). A technique for using neural network analysis to perform survival analysis of censored data. *Cancer Letters, 77,* 127-138.

Engelbrecht, A.P., Cloete, I. y Zurada, J.M. (1995). Determining the significance of input parameters using sensitivity analysis. En J. Mira y F. Sandoval (Eds.), *Proceedings of International Workshop on Artificial Neural Networks* (pp. 382-388). New York: Springer.

Frost, F. y Karri, V. (1999). Determining the influence of input parameters on BP neural network output error using sensitivity analysis. En B. Verma, H. Selvaraj, A. Carvalho y X. Yao (Eds.), *Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications* (pp.45-49). Los Alamitos, CA: IEEE Computer Society Press.

Fu, L. y Chen, T. (1993). Sensitivity analysis for input vector in multilayer feedforward neural networks. En IEEE (Ed.), *Proceedings of IEEE International Conference on Neural Networks* (pp. 215-218). New York: IEEE.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks, 2,* 183-192.

Garson, G.D. (1991a). Interpreting neural-network connection weights. *AI Expert, April,* 47-51.

Garson, G.D. (1991b). A comparison of neural network and expert systems algorithms with common multivariate procedures for analysis of social science data. *Social Science Computer Review, 9(3),* 399-434.

Gedeon, T.D. (1997). Data mining of inputs: analysing magnitude and functional measures. *International Journal of Neural Systems, 8(2),* 209-218.

Guo, Z. y Uhrig, R.E. (1992). Sensitivity analysis and applications to nuclear power plant. En IEEE (Ed.), *International Joint Conference on Neural Networks* (pp. 453-458). Piscataway, NJ: IEEE.

Harrison, R.F., Marshall, J.M. y Kennedy, R.L. (1991). The early diagnosis of heart attacks: a neurocomputational approach. En IEEE (Ed.), *Proceedings of IEEE International conference on Neural Networks* (pp. 231-239). New York: IEEE.

Hashem, S. (1992). Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions. En IEEE (Ed.), *International Joint Conference on Neural Networks* (pp. 419-424). New York: IEEE.

Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2(5),* 359-366.

Hunter, A., Kennedy, L., Henry, J. y Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine, 62,* 11-19.

Hwang, J.N., Choi, J.J., Oh, S. y Marks, R.J. (1991). Query based learning applied to partially trained multilayer perceptron. *IEEE Transactions on Neural Networks, 2(1),* 131-136.

Kashani, J.H., Nair, S.S., Rao, V.G., Nair, J. y Reid, J.C. (1996). Relationship of personality, environmental, and DICA variables to adolescent hopelessness: a neural network sensitivity approach. *Journal of American Children and Adolescent Psychiatry, 35(5),* 640-645.

Lisboa, P.J.G., Mehridehnavi, A.R. y Martin, P.A. (1994). The interpretation of supervised neural networks. En P.J.G. Lisboa y M.J. Taylor (Eds.), *Proceedings of the Workshop on Neural Network Applications and Tools* (pp. 11-17). Los Alamitos, CA: IEEE Computer Society Press.

Masters, T. (1993). *Practical neural networks recipes in C++.* London: Academic Press.

Milne, K. (1995). Feature selection using neural networks with contribution measures. En IEEE (Ed.), *Proceedings of Australian Conference of Artificial Intelligence* (pp. 124-136). Sydney: IEEE West Australian Section.

Modai, I., Saban, N.I., Stoler, M., Valevski, A. y Saban, N. (1995). Sensitivity profile of 41 psychiatric parameters determined by neural network in relation to 8-week outcome. *Computers in Human Behavior, 11(2),* 181-190.

Opara, J., Primozic, S. y Cvelbar, P. (1999). Prediction of pharmacokinetic parameters and the assessment of their variability in bioequivalence studies by artificial neural networks. *Pharmaceutical Research, 16(6),* 944-948.

Rambhia, A.H., Glenny, R. y Hwang, J. (1999). Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis. En IEEE (Ed.), *IEEE International Conference on Acoustics,* Speech, and Signal Processing (pp. 1097-1100). Piscataway, NJ: IEEE.

Reid, J.C., Nair, S.S., Kashani, J.H. y Rao, V.G. (1994). Detecting dysfunctional behavior in adolescents: the examination of relationships using neural networks. En P.W. Lefley (Ed.), *Proceedings of Annual Symposium of Computational Applications on Medical Care* (pp. 743-746). New York: Springer.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En D.E. Rumelhart y J.L. McClelland (Eds.), *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet.* New York: Springer-Verlag.

Sarle, W.S. (Ed.) (1998). *Neural network FAQ.* Recuperado 20/10/01, desde la dirección Internet <ftp://ftp.sas.com/pub/neural/FAQ.html>.

Sarle, W.S. (2000). *How to measure importance of inputs?* Recuperado 2/11/01, desde la dirección Internet <ftp://ftp.sas.com/pub/neural/importance.html>.

Takenaga, H., Abe, S., Takatoo, M., Kayama, M., Kitamura, T. y Okuyama, Y. (1991). Input layer optimization of neural networks by sensitivity analysis and its application to recognition of numerals. *Transactions of the Institute of Electrical Engineers Japan,* 111(1), 36-44.

Tsaih, R. (1999). Sensitivity analysis, neural networks, and the finance. En IEEE (Ed.), *International Joint Conference on Neural Networks* (pp. 3830-3835). Piscataway, NJ: IEEE.

Yoon, Y.O., Brobst, R.W., Bergstresser, P.R. y Peterson, L.L. (1989). A desktop neural network for dermatology diagnosis. *Journal of Neural Network Computing, 1,* 43-52.

Yoon, Y., Swales, G. y Margavio, T.M. (1993). A comparison of discriminant analysis versus artificial neural networks. *Journal of the Operational Research Society, 44(1),* 51-60.

Zurada, J.M., Malinowski, A. y Cloete, I. (1994). Sensitivity analysis for minimization of input data dimension for feedforward neural network. En IEEE (Ed.), *Proceedings of IEEE International Symposium on Circuits and Systems* (pp. 447-450). New York: IEEE.

Original recibido: 21/1/2002
Versión final aceptada: 2/5/2002

2.6.
Numeric sensitivity analysis applied to feedforward neural networks.

Neural Comput & Applic (2003) 12: 119–125
DOI 10.1007/s00521-003-0377-9

ORIGINAL ARTICLE

J. J. Montaño · A. Palmer

Numeric sensitivity analysis applied to feedforward neural networks

Received: 2 May 2002 / Accepted: 16 May 2003

Abstract During the last 10 years different interpretative methods for analysing the effect or importance of input variables on the output of a feedforward neural network have been proposed. These methods can be grouped into two sets: analysis based on the magnitude of weights; and sensitivity analysis. However, as described throughout this study, these methods present a series of limitations. We have defined and validated a new method, called Numeric Sensitivity Analysis (NSA), that overcomes these limitations, proving to be the procedure that, in general terms, best describes the effect or importance of the input variables on the output, independently of the nature (quantitative or discrete) of the variables included. The interpretative methods used in this study are implemented in the software program *Sensitivity Neural Network 1.0*, created by our team.

Keywords Neural networks · Sensitivity analysis · Input impact

1 Introduction

In Artificial Neural Networks (ANN) research, most efforts have centred on the development of new learning rules, the exploration of new neural network architectures and the expansion of new fields of application. Not much attention has been dedicated to the development of procedures that would permit the understanding of the nature of the internal representations generated by the network to respond to a given problem. Instead, ANNs have been presented to the user as a kind of 'black box' whose extremely complex work transforms inputs into predetermined outputs. In other words, it is not possible to find out immediately how the weights of the network or the activation values of the hidden neurons are related to the set of data being handled. Thus, unlike classic statistical models, in a network it does not appear to be easy to find out the effect that each explicative variable has on the dependent variable.

Since the end of the 1980s, different methods have been proposed for interpreting what has been learned by a feedforward neural network composed of input neurons N, hidden neurons L, and output neurons M. As shown in Fig. 1, these interpretative methods can be divided in two types of methodologies: analysis based on the magnitude of weights; and sensitivity analysis.

Analysis based on the magnitude of weights groups together those procedures that are based exclusively on the values stored in the static matrix of weights to determine the relative influence of each input variable on each one of the network outputs. Different equations have been proposed based on the weights magnitude [1–7], all of them characterised by the calculation of the product of the weights w_{ij} (connection weight between the input neuron i and the hidden neuron j) and v_{jk} (connection weight between the hidden neuron j and the output neuron k) for each of the hidden neurons of the network, obtaining the sum of the calculated products. The equation below is that proposed by Garson [3] as representative of this type of analysis:

$$Q_{ik} = \frac{\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)}{\sum_{i=1}^{N}\left(\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)\right)} \quad (1)$$

where $\sum_{r=1}^{N} w_{rj}$ is the sum of the connection weights between the N input neurons and the hidden neuron j, and Q_{ik} represents the percentage of influence of the input variable x_i on the output y_k, in relation to the rest of the input variables in such a way that the sum of this index must give a value of 100% for all of the input variables.

J. J. Montaño (✉) · A. Palmer
Facultad de Psicología. Universidad de las Islas Baleares,
Ctra. Valldemossa Km. 7,5, 07122 Palma de Mallorca, Spain
E-mail: {juanjo.montano, alfonso.palmer}@uib.es

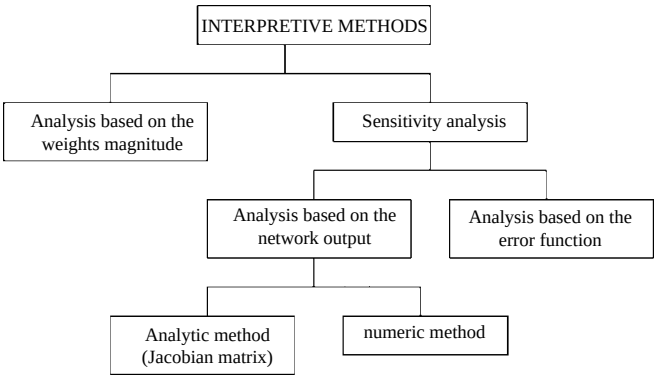


Fig. 1 Scheme of the proposed interpretative methods

Tchaban et al [8] recently presented an interesting variant on this type of analysis, called 'weight product', that incorporates the ratio of the value of the input variable x_i and the value of the output y_k to the product of the weights by means of the expression

$$WP_{ik} = \frac{x_i}{y_k} \sum_{j=1}^{L} w_{ij} v_{jk} \tag{2}$$

where WP_{ik} represents the influence of the input variable x_i on the output y_k.

Different empirical studies have demonstrated that analysis based on the magnitude of weights is not effective for determining the effect of the input variables on the output of a neural network [3, 6, 9, 10].

Sensitivity analysis is based on the measurement of the effect that is observed in the output y_k due to the change that is produced in the input x_i. Thus, the greater the effect observed in the output, the greater the sensitivity present with respect to the input. Obtaining the Jacobian matrix by the calculation of the partial derivatives of the output y_k with respect to the input x_i, that is $\frac{\partial y_k}{\partial x_i}$, constitutes the analytical version of sensitivity analysis [11, 12]. By applying the chain rule to $\frac{\partial y_k}{\partial x_i}$ we have:

$$S_{ik} = \frac{\partial y_k}{\partial x_i} = f'(net_k) \sum_{j=1}^{L} v_{jk} f'(net_j) w_{ij} \tag{3}$$

where S_{ik} represents the sensitivity of the output y_k due to changes in the input variable x_i, $f'(net_j)$ and $f'(net_k)$ are the derivative of the activation function of the hidden neuron j and the output neuron k, respectively.

The calculation of the partial derivatives is quite useful in determining the importance of the input because it represents the instant slope of the underlying function between each pair of input x_i and output y_k.

As can be seen, the values for the Jacobian matrix do not depend only on the information learned by the neural network, which is stored in a distributed way in the connections w_{ij} and v_{jk}. Rather, they also depend upon the activation of the neurons in the hidden layer and the output layer, which, in turn, depend on the input patterns. Since different input patterns can provide different values for the slope, the sensitivity is generally found by calculating the arithmetic mean and the standard deviation of S_{ik} obtained from the training data set. The same procedure is followed in the weight product method, due to the fact that each input pattern provides a different WP_{ik} value.

The application of Analytic Sensitivity Analysis (ASA) has been very useful in such diverse fields as image recognition [13], engineering [14, 15], meteorology [16] and medicine [17–19]. Nevertheless, this method is limited as far as its number of potential applications, because it is based on the assumption that all of the variables included in the model are quantitative [9]. When using discrete variables (binary, for example), the partial derivative does not seem to provide any practical significance.

Sensitivity analysis can also be applied to the effect observed in a determined error function, provoking a change or perturbation in the input. A common way to carry out this type of analysis, called 'clamping technique' [10], consists of comparing the error made by the network from the original patterns with the error made when restricting the input of interest to a fixed value (in general the average value) for all patterns. Thus, the greater the increase in the error function upon restricting the input, the greater the importance of this input in the output. As in the previous case, this method is quite useful when applied to quantitative input variables, however, it is not easily applied to discrete variables since it would not be correct to attach a variable of a discrete nature (yes = 1 or no = 2, for example) to an average value.

With the aim of overcoming the limitations observed in the methods that have been proposed to date, we have developed a new method called Numeric Sensitivity Analysis (NSA), based on the calculation of the slopes that are formed between the inputs and the outputs, without making any assumptions about the nature of the variables included.

2 Numeric sensitivity analysis

To analyse the effect of an input variable x_i on an output variable y_k by means of NSA method we must first arrange the patterns p in ascending order according to the values of the input variable x_i. In function of this order a determined number G of groups of equal, or approximately equal, size are generated. The ideal number of groups depends on the number of available patterns and the complexity of the function that is established between the input and the output, although in most cases a value of $G = 30$, or a similar value, is sufficient. For each group g_r formed, the arithmetic mean of the variable x_i and the arithmetic mean of the variable y_k is calculated. Then the NSA index is obtained based on the numeric calculation of the slope formed between each pair of

consecutive groups g_r and g_{r+1}, of x_i over y_k by means of the following expression:

$$NSA_{ik}(g_r) \equiv \frac{\bar{y}_k(g_{r+1}) - \bar{y}_k(g_r)}{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)} \quad (4)$$

where $\bar{x}_i(g_r)$ and $\bar{x}_i(g_{r+1})$ are the means of the variable x_i corresponding to the groups g_r and g_{r+1}, respectively and $\bar{y}_k(g_r)$ and $\bar{y}_k(g_{r+1})$ are the means of the variable y_k corresponding to the groups g_r and g_{r+1}, respectively.

Once the *NSA G-1* values are calculated, the value of the expected value of the *NSA* index or slope between the input variable x_i and the output variable y_k can be obtained by means of:

$$E(NSA_{ik}(g_r)) = \sum_{r=1}^{G-1} NSA_{ik}(g_r) f(NSA_{ik}(g_r)) = \frac{\bar{y}_k(g_G) - \bar{y}_k(g_1)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)} \quad (5)$$

where $f(NSA_{ik}(g_r)) \equiv \frac{\bar{x}_i(g_{r+1}) - \bar{x}_i(g_r)}{\bar{x}_i(g_G) - \bar{x}_i(g_1)}$ represents the probability function of the *NSA* index, $\bar{x}_i(g_G)$ and $\bar{x}_i(g_1)$ are the average values of the variable x_i for the last group g_G and the first group g_1, respectively and $\bar{y}_k(g_G)$ and $\bar{y}_k(g_1)$ are the average values of the variable y_k for the group g_G and the group g_1, respectively.

When the input variable x_i is binary discrete (values 0 and 1, for example) the expected value of the *NSA* index is obtained by calculating the mean of the variable y_k when the variable x_i takes the minimum value, and the mean of the variable y_k when the variable x_i takes the maximum value, and by applying the expression

$$E(NSA_{ik}(g_r)) = \frac{\bar{y}_k(x_{i\max}) - \bar{y}_k(x_{i\min})}{x_{i\max} - x_{i\min}} \quad (6)$$

where $x_{i\max}$ and $x_{i\min}$ are the maximum and minimum values respectively that the variable x_i can take, and $\bar{y}_k(x_{i\max})$ and $\bar{y}_k(x_{i\min})$ are the mean of the variable y_k when the variable x_i takes the maximum value, and the mean of the variable y_k when the variable x_i takes the minimum value, respectively.

The expected value of the *NSA* index represents the average effect that an increment of x_i has over y_k. When the input variable x_i is binary, the expected value represents the average effect provoked by the change of the minimum value to the maximum value in the variable x_i.

In the application of this sensitivity analysis method, we must take into account, on the one hand, that the categorical variables should be represented by dummy binary variables (dummy coding) and, on the other, the input and output variables should be re-scaled to the same range of possible values (between 0 and 1, for example). This avoids possible biases in obtaining the *NSA* index due to the use of different measurement scales between input variables, and also allows us to obtain a standardised value of expected value, unlike in the ASA method. Thus, the range of possible values that it can adopt $E(NSA_{ik}(g_r))$, oscillates between -1 and $+1$. These two limits indicate a maximum effect of the input variable on the output, with a negative relation in the first case (-1) and a positive relation in the second case ($+1$). The values near or equal to zero indicate the absence of effect of the input variable.

The calculation of the standard deviation of the *NSA* index, when the input variable x_i is quantitative, can be done by:

$$SD(NSA_{ik}(g_r)) = \sqrt{E(NSA_{ik}^2(g_r)) - (E(NSA_{ik}(g_r)))^2} \quad (7)$$

The value of the standard deviation should be interpreted as the quantity of oscillations that the slope has undergone, which is established between x_i and y_k, in such a way that the greater the value of the standard deviation, the more chaotic or random the behaviour of the function between the two variables included.

When the input variable is binary discrete, the standard deviation is not calculable because in this case the value of $E(NSA_{ik}(g_r))$ is obtained by the calculation of only one slope (expression (6)), and, as has already been stated, with the NSA method the standard deviation represents the magnitude of the oscillations undergone by the different slopes calculated between the input variable x_i and the output variable y_k.

Finally, it is opportune to show the graphic representation of the slope between each pair of input-output variables together with the expected value and standard deviation indexes. This is especially useful in those cases in which the value of the expected value could mask the underlying function between the input variable and the output variable.

3 Comparative study

3.1 Description

The purpose of this study is to carry out a comparative study of the yield of the interpretative methods described above: Garson's, weight product, ASA and NSA. Therefore, four data matrixes were generated by means of simulation, each one composed of 1000 registers and four variables with a range between 0 and 1. The first three variables (X1, X2 and X3) act as predictor variables or input variables in the network, while the last variable (Y) is a function of the predictor variables and acts as an output variable. The value of Pearson's correlation coefficient between the predictor variables oscillates between 0 and 0.35. In all cases the variable X1 has no contribution or effect on the output Y, followed by the variable X2 with an intermediate effect and the variable X3 that shows the greatest effect on the network output. With the aim of analysing the behaviour of the different methods depending on the type of variable included, the nature of the input variables were manipulated in each of the matrixes. A detailed description of each matrix used is presented below:

Matrix 1: Quantitative variables

The input variables are of a quantitative nature with normal distribution. To generate the Y variable as a function of the input variables the following expression was used:

$$Y = 0.0183\exp(4X2) + \tanh(X3) + ErrorN(0, 0.1) \quad (8)$$

In this way, X1 does not have any contribution in the Y variable. The function that is established between X2 and the output variable is of the exponential type, with a range between 0 and 1 for the output variable, while the function between X3 and the output variable is the hyperbolic tangent with a range between −1 and 1 for the output variable. Finally, a random error was added with a normal distribution, 0 mean and a standard deviation of 0.1. The resulting output variable was re-scaled to values between 0 and 1.

Matrix 2: Discrete binary variables

The input and output variables are of a discrete binary nature with values 0 and 1. The relation between the inputs and the output was determined by Phi coefficient. Thus, the values of the Phi coefficient between X1, X2, X3 and the output were 0.013, 0.378 and 0.606, respectively.

Matrix 3: Discrete variables with multiple categories

The input variables are of a discrete nature with values 1, 2 and 3, while the output variable is discrete binary, with values 0 and 1. The relation between the inputs and the output was determined by V coefficient. Thus, the values of the V coefficient between X1, X2, X3 and the output were 0.066, 0.418, and 0.636, respectively. Each predictor variable was codified by means of the use of two dummy binary variables (value $1 = 0\ 0$, value $2 = 1\ 0$ and value $3 = 0\ 1$), in such a way that each variable was represented by two input neurons.

Matrix 4: Quantitative and discrete variables

The X1 variable is of a discrete nature with values 1, 2 and 3 codified by means of two dummy variables, the X2 variable is discrete binary with values 0, 1 and the X3 variable is quantitative with normal distribution. To generate the Y variable as a function of the input variables the following expression was used:

$$\begin{cases} \text{If } X2 = 1, \text{ then } Y = 0.75 + 2.5\exp(-1.5X3) \\ \qquad\qquad + Error\ N(0, 0.1) \\ \text{If } X2 = 0, \text{ then } Y = 2.5\exp(-1.5X3) \\ \qquad\qquad + Error\ N(0, 0.1) \end{cases} \quad (9)$$

In this way, X1 does not contribute in the Y variable. Also, when X2 takes the value of 1, the Y variable is incremented in 0.75 units. The function that is established between X3 and the Y variable is of the negative exponential type, with a range between 0.5 and 2.5 for the output variable. As in the case of matrix 1, a random error was added with normal distribution, 0 mean and a standard deviation of 0.1. The resulting output variable was re-scaled to values between 0 and 1.

To carry out this study, the software program *Sensitivity Neural Network 1.0* was used. This program, created by the authors, permits, through an easy to use interface, the simulation of the behaviour of a feedforward neural network trained with the backpropagation learning rule, and incorporates for the first time the described interpretative methods. To train the neural networks each data matrix was divided into three groups: 500 patterns acted as a training set, 250 patterns acted as a validation set and 250 patterns acted as a test set. The simulated neural networks were composed of three, four or six input neurons depending on the data matrix, two hidden neurons and an output neuron. As activation functions, the hyperbolic tangent was used in the hidden neurons, and the linear function was used in the output neuron. As learning parameters, a value of 0.25 was used for the learning rate, and a value of 0.8 for the momentum factor. The connection weights and threshold weights were initialised with different seed values. The neural network that obtained the best results taking into account the validation set of the corresponding matrix, was selected to go on to the test phase. The four neural models that were finally obtained showed a good fit with the test data.

3.2 Results

Table 1 shows the results obtained after applying the interpretative methods to the neural models selected in the validation phase. The sensitivity analyses (analytic and numeric) and the weight product method were calculated using the training set as data, while Garson's method was applied only to the connection weights of the network.

It can be observed that, in general, the four methods analysed correctly establish the hierarchy among the input variables in function of their importance.

A more detailed analysis indicates that the four methods behave correctly when the input variables are quantitative (matrix 1), and when they are quantitative and discrete (matrix 4). In this case, the NSA and ASA methods obtain very similar average values.

When the input variables are discrete binary (matrix 2), it can be observed that the NSA method is the one that mostly closely represents reality, because this index coincides with the real values of the Phi coefficient between the input variables and the network output. It can also be observed that the ASA method, in spite of being the one that provides the average values

Table 1 Results obtained by applying the interpretative methods to the neural models

Interpretive methods							
	Garson's method	Weight Product		Analytic Sensitivity analysis		Numeric Sensitivity analysis	
		Mean	S.D.	mean	S.D.	mean	S.D.
Matrix 1							
X1	1.395	0.009	0.025	0.005	0.003	0.035	2.855
X2	23.270	0.374	0.366	0.199	0.038	0.143	2.600
X3	75.340	3.194	3.906	1.376	0.665	0.977	0.877
Matrix 2							
X1	9.293	−0.254	0.647	−0.016	0.008	0.013	–
X2	32.770	0.378	0.438	0.338	0.091	0.378	–
X3	57.940	0.450	0.458	0.527	0.138	0.606	–
Matrix 3							
X1							
X1V1	6.139	−0.343	3.387	0.050	0.080	0.057	–
X1V2	6.065	−0.100	2.345	0.038	0.073	0.006	–
Sum	12.204	−0.443		0.088		0.063	
X2							
X2V1	14.660	2.371	6.277	0.278	0.273	0.040	–
X2V2	24.990	0.669	1.183	0.486	0.473	0.361	–
Sum	39.650	3.040		0.7640		0.401	
X3							
X3V1	12.290	1.334	2.940	0.315	0.272	−0.020	–
X3V2	35.860	0.582	0.850	0.627	0.642	0.591	–
Sum	48.150	1.916		0.942		0.611	
Matrix 4							
X1							
X1V1	0.791	−0.004	0.007	−0.001	0.001	−0.001	–
X1V2	0.122	−0.001	0.001	−0.001	0.001	−0.010	–
Sum	0.913	−0.005		−0.002		−0.011	
X2	28.000	0.293	0.311	0.260	0.054	0.268	–
X3	71.080	−2.201	2.057	−0.648	0.174	−0.643	1.465

nearest to the NSA method, shows a clear discrepancy with respect to the real value. This aspect can be better appreciated by observing Table 2, where the Phi values and the indexes of the NSA and ASA methods for matrix 2 are shown. As Table 1 indicates, Garson's method and the weight product method overrate the importance of X1 giving a value of 9.29% and −0.254, respectively.

When the input variables are discrete with multiple categories (matrix 3), we can verify that, again, the NSA method is the one that best fits with reality, because it is the method that is closest to the values of V coefficient. In order to verify this, as shown in Table 1, the sum of the absolute values obtained with the two dummy variables that represent each predictor variable was calculated. Table 2 allows us to compare the real value of the V coefficient with the indexes of the NSA and ASA methods for matrix 3. As in the previous case, Garson's method and the weight product method overrate the importance of X1, providing a value of 12.204% and −0.443, respectively. The weight product method provides, with this data matrix, higher values for X2 (3.040) than for X3 (1.916) and, therefore, does not correctly establish the hierarchy of importance among the input variables.

With respect to the reliability of the different methods, the ASA method is the one that is the most accurate in estimating the mean value (note the low standard deviations). The weight product method is the least accurate, obtaining very high values of standard deviation in some cases. Note that in the case of the NSA method, the value of the standard deviation should be interpreted, contrary to the ASA and weight product methods, as the degree of oscillation that the slope undergoes which is established between a quantitative input and the output. Thus, Table 1 shows that in matrix 1, the more randomness existing in the function that is established between the two variables included, the greater the standard deviation value of the slope.

With the NSA method, the interpretation of the effect of a variable is much simpler, since the mean that it provides is restricted to values between −1 and 1. Note that this condition is fulfilled when the variables are restricted to a same range of values. The weight

Table 2 Values for the Phi and V coefficients and indexes obtained by means of the ASA and NSA methods for the matrices 2 and 3

	X1	X2	X3
Matrix 2			
Phi coeff.	0.013	0.378	0.606
Analytic SA	−0.016	0.338	0.527
Numeric SA	0.013	0.378	0.606
Matrix 3			
V coeff.	0.066	0.418	0.636
Analytic SA	0.088	0.764	0.942
Numeric SA	0.063	0.401	0.611

product method and the ASA method however, provide average values that can virtually oscillate between $-\infty$ y $+\infty$

Finally, Fig. 2 shows the graphic representation of the NSA method obtained by the program *Sensitivity Neural Network 1.0* for matrix 1. It can be seen that the neural network correctly learned the underlying function between the input variables and the output. Thus, we can derive that X1 does not have a systematic effect on the output, X2 maintains a slight exponential function and X3 maintains a clear sigmoid function with the output.

4 Summary and discussion

A comparative study on the usefulness of four methods was carried out, aimed at evaluating the relative importance of the input variables in a feedforward neural network. For this, four data matrices were obtained by means of simulation, in which both the nature and the degree of the relation or importance between the variables were manipulated. From these results we can infer a series of advantages in the application of the NSA method presented in this paper with respect to the methods previously proposed.

In the first place, the NSA is the method which, in general terms, allows for the best description of the effect or importance of the input variables on the output. It has been possible to observe that the NSA and ASA methods provide similar average values when the variables included are quantitative. The ASA method is perfectly valid in these cases, even taking into account the low variability that it provides. In this sense, previous studies corroborate this conclusion [13–19].

However, when the variables included are discrete (binary and discrete with multiple categories), the NSA method is the most suitable because the values that it provides coincide with the Phi index in the case of binary variables, and are quite close to V index in the case of variables with multiple categories. These two indexes are the ones that, in the field of statistics, have been used to analyse the relation between discrete variables in two-way contingency tables. The Phi index is especially useful when the variables included are binary, while V index can be used both in the case of binary variables and variables with multiple categories.

Garson's method and the weight product method in some cases overrate the importance of variables that are irrelevant to the network output. Coinciding with the results obtained by Wang et al [10], the weight product method is not reliable, judging by the values of standard deviation that it provides.

Secondly, with the NSA method the interpretation of the effect of a variable is much simpler, since the *NSA* index that it provides is restricted to the interval (−1, 1) unlike with the weight product and ASA methods.

Finally, NSA method incorporates a procedure that allows for the graphic representation of the function learned by the network between an input variable and the output of the network. This graphic representation shows relevant information that complements the

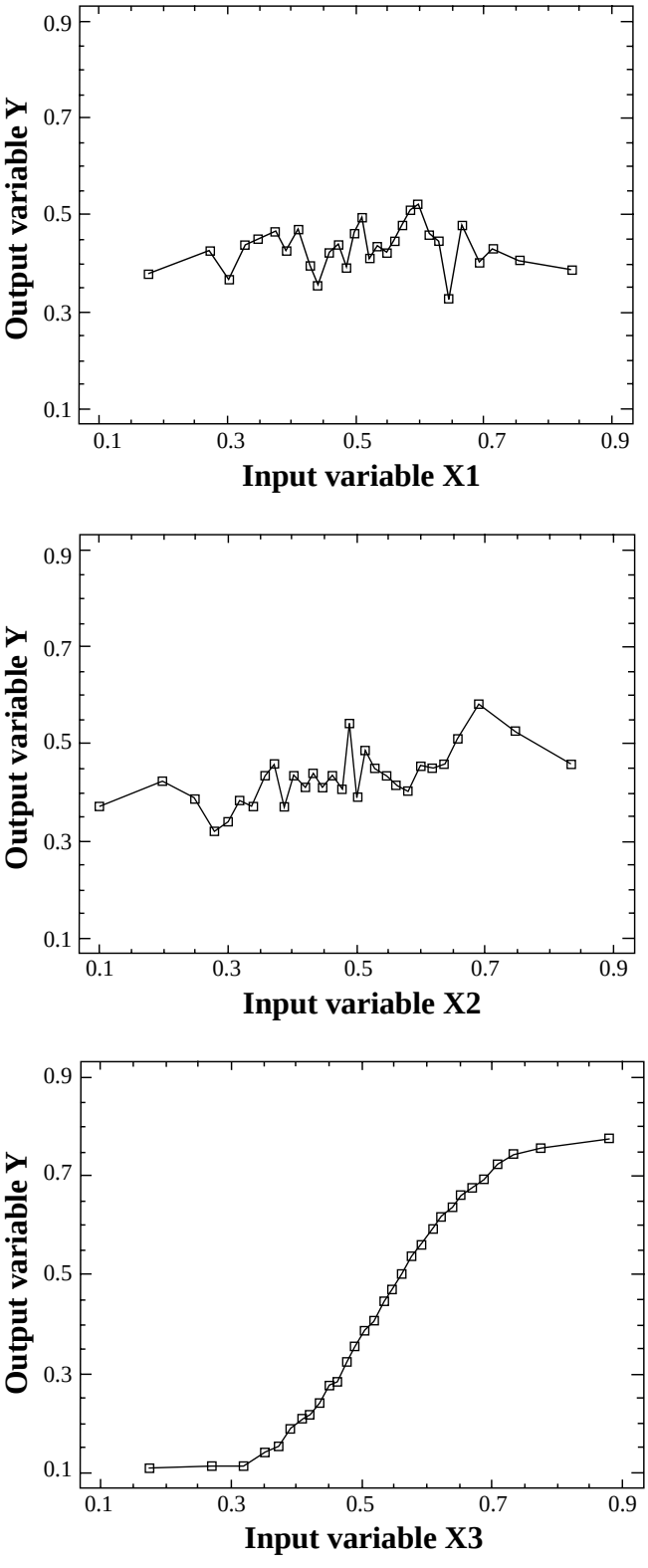


Fig. 2 Graphic representation of numeric sensitivity analysis applied to the data matrix 1

information provided by the numerical indexes, because, in many cases, a summary index is not enough to reflect the underlying function between variables.

In summary, it can be said that the NSA method is the one that permits a more precise evaluation of the importance or effect of the input variables, independently of their nature (quantitative or discrete), thus overcoming the limitations of the methods that have been proposed up until the present. The number of fields of potential application where the use of discrete variables is very common: medicine, sociology, psychology, biology, economics, etc. is considerably increased with this method. Now we can not only construct potent instruments of prediction by means of feedforward neural networks, but we can also analyse the impact of the predictor variables in any set of data from the networks.

Future work on this project will focus on replicating the results obtained by using simulated matrices that establish different scenarios from those that were used in this study. Thus, one would be able to observe the effect of manipulating the degree of relation between the input variables or the type of relation (linear or non-linear) between inputs and outputs on the efficacy of the proposed methods. The practical efficacy of these methods should also be tested on real data matrices.

The NSA method, together with the previously proposed methods, is implemented in the software program *Sensitivity Neural Network 1.0*, created by our team. The user can analyse the qualities of each one of the interpretative methods with respect to the others for different configurations of data. This program is available to the public, and can be obtained by contacting the authors via e-mail.

References

1. Yoon YO, Brobst RW, Bergstresser PR, Peterson LL (1989) A desktop neural network for dermatology diagnosis. J Neural Network Computing 1:43–52
2. Baba K, Enbutu I, Yoda M (1990) Explicit representation of knowledge acquired from plant historical data using neural network. In: Proceedings International Conference on Neural Networks, IEEE, New York, pp. 155–160
3. Garson GD (1991) Interpreting neural-network connection weights. AI Expert 47–51
4. Yoon Y, Swales G, Margavio TM (1993) A comparison of discriminant analysis versus artificial neural networks. J Operational Research Society 44:51–60
5. Milne K (1995) Feature selection using neural networks with contribution measures. In: Proceedings Australian Conference AI'95, IEEE West Australian Section, Sydney, pp. 124–136
6. Gedeon TD (1997) Data mining of inputs: analysing magnitude and functional measures. International J Neural Systems 8:209–218
7. Tsaih R (1999) Sensitivity analysis, neural networks, and the finance. In: IJCNN'99, IEEE, Piscataway, NJ, pp. 3830–3835
8. Tchaban T, Taylor MJ, Griffin A (1998) Establishing impacts of the inputs in a feedforward network. Neural Computing & Applications 7:309–317
9. Sarle WS (2000) How to measure importance of inputs? ftp://ftp.sas.com/pub/neural/importance.html
10. Wang W, Jones P, Partridge D (2000) Assessing the impact of input features in a feedforward neural network. Neural Computing & Applications 9:101–112
11. Zurada JM, Malinowski A, Cloete I (1994) Sensitivity analysis for minimization of input data dimension for feedforward neural network. In: Proceedings IEEE International Symposium on Circuits and Systems, IEEE, New York, pp. 447–450
12. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
13. Takenaga H, Abe S, Takatoo M, Kayama M, Kitamura T, Okuyama Y (1991) Input layer optimization of neural networks by sensitivity analysis and its application to recognition of numerals. Transactions of the Institute of Electrical Engineers Japan 111:36–44
14. Guo Z, Uhrig RE (1992) Sensitivity analysis and applications to nuclear power plant. In: International Conference on Neural Networks, IEEE, Piscataway, NJ, pp. 453–458
15. Bahbah AG, Girgis AA (1999) Input feature selection for real-time transient stability assessment for artificial neural network (ANN) using ANN sensitivity analysis. In: Proceedings of PICA'99, IEEE, Piscataway, NJ, pp. 295–300
16. Castellanos J, Pazos A, Ríos J, Zafra JL (1994) Sensitivity analysis on neural networks for meteorological variable forecasting. In: Vlontzos J, Hwang JN, Wilson E (eds) Proceedings IEEE – WNNSP, IEEE, New York, pp. 587–595
17. Harrison RF, Marshall JM, Kennedy RL (1991) The early diagnosis of heart attacks: a neurocomputational approach. In: Proceedings IEEE - IJCNN'91, IEEE, New York, pp. 231–239
18. Engelbrecht AP, Cloete I, Zurada JM (1995) Determining the significance of input parameters using sensitivity analysis. In: Mira J, Sandoval F (eds) Proceedings IWANN, Springer, New York, pp. 382–388
19. Rambhia AH, Glenny R, Hwang J (1999) Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis. In: Proceedings ICASSP99, IEEE, Piscataway, NJ, pp. 1097–1100

2.7.
Sensitivity neural network: an artificial neural network simulator with sensitivity analysis.

Behavior Research: Methods, Instruments, & Computers (en revisión)

Sensitivity Neural Network: an artificial neural network simulator with sensitivity analysis

Alfonso Palmer Pol Juan José Montaño Moreno
Universidad de las Islas Baleares

Carlos Fernández Provencio
InfoMallorca, S.L.

Abstract

This article presents the Sensitivity Neural Network program which permits the simulation of the behavior of a multilayer perceptron neural network (with one input layer, one hidden layer and one output layer) trained by means of the backpropagation error learning rule and incorporating for the first time diverse methods – generically called sensitivity analysis -, that, in literature on neural networks, have shown to be useful for the study of the importance or effect of each input variable on the network output: Garson's method based on the magnitude of the connection weights, analytic sensitivity analysis based on the computation of the Jacobian matrix and the numeric sensitivity analysis proposed by the authors. Sensitivity Neural Network has a user friendly interface, works under the operating system Windows and is distributed without cost.

Artificial Neural Networks (ANN) of the multilayer perceptron type associated with the backpropagation error learning rule (Rumelhart, Hinton, & Williams, 1986) have been the most widely used in the field. Specifically, the multilayer perceptron has been used as a tool for prediction, oriented principally towards pattern classification and estimation of continuous variables, obtaining excellent results compared with classic statistical models. However, one of the most important criticisms of neural networks cites how difficult it is to understand the nature of the internal representations generated by the network. Unlike classic statistical models, in a neural network it is not so easy to find out the importance or effect that each input variable has on the output. Thus, ANNs have been presented to the user as a kind of "black box" whose extremely complex work somehow magically transforms inputs into predetermined outputs.

With the aim of overcoming this limitation, different numeric methods have been proposed in an attempt to determine what has been learned by the neural network,

obtaining very promising results. Amongst these methods we can single out Garson's method (1991) as representative of analysis based on the magnitude of weights, the analytic sensitivity analysis based on the computation of the Jacobian matrix (Bishop, 1995) and the numeric sensitivity analysis developed recently by us, which supposes an improvement regarding the previous methods. In spite of the usefulness of these methods, there is still no software available which could implement them.

In this paper we present the program Sensitivity Neural Network, which allows us to simulate the behavior of a multilayer perceptron network (with an input layer, a hidden layer, and an output layer) trained by means of the backpropagation error learning rule and which, for the first time, incorporates the aforementioned group of interpretative methods – Garson's method, analytic sensitivity analysis and numeric sensitivity analysis – directed at determining the importance or the effect of the input variables on the network output. Complementary to these numerical methods, the program also incorporates a procedure for visualization, showing the graphic representation of the underlying function that the network has learned between every pair of input-output variables.

As shown in figure 1 Sensitivity Neural Network is composed of a principal window, which is divided into a series of sections that easily permit the manipulation of the most relevant aspects of the training and validation of a multilayer perceptron network. Thus the section *Data files* allows us to select and visualize on one spread sheet the matrices of data that will be used for the training, validation, and testing of the neural model. In the section *Network topology* you can configurate the number of neurons in the hidden layer, the type of activation function (linear, logistic sigmoidal and hyperbolic tangent sigmoidal) of the hidden and output neurons, as well as determine the value of the learning rate and the momentum factor. The section *Weights* permits the determination of inicial connection and threshold weights of the network by means of a random generation process, or by importing the weights obtained in a previous session or by the use of another simulator program. In the section *Stop criteria* you determine the stop criteria of the training, which can be in function of the yield of the network in the presence of the training and validation data, or in function of a predetermined number of epochs in the training. During the process of training the network, the section *Statistics* reveals a series of statistical indexes together with a graphic representation that describe the yield of the neural model.

Figure 1. Principal window of the Sensitivity Neural Network program.

Once a configuration of optimal weights has been obtained from a data set you can ask the program for a report of the results. The report (see figure 2) gives information about the network topology, the value of the connection and threshold weights obtained, the output desired by the user and the output estimated by the network for each pattern that makes up the data matrix, the global evaluation of the model regarding the value of the mean square of error and the number of patterns correctly classified, as well as the results of the analysis of the input effect by means of the interpretative methods. Finally, Sensitivity Neural Network offers a graphic representation of the underlying function learned by the network between every pair of input-output variables.

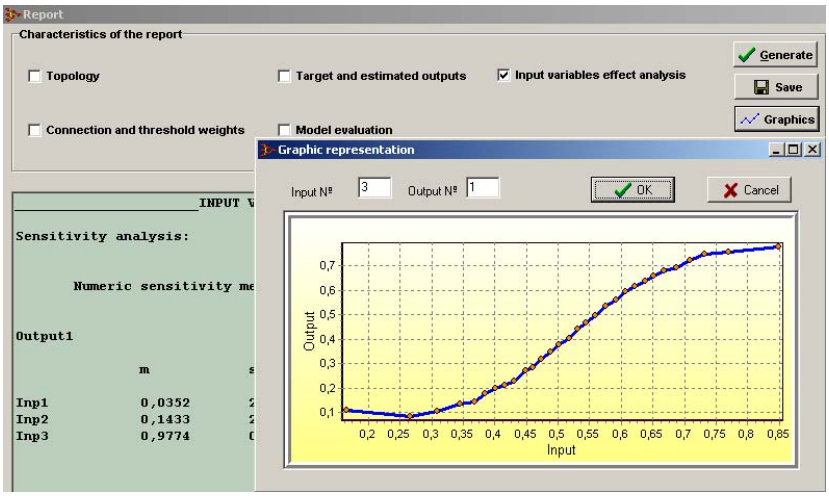


Figure 2. Window showing the report of the results of the Sensitivity Neural Network program.

Example

We generated, by simulation, a data matrix composed of 500 patterns and three variables of a continuous nature with normal distribution in the range (0,1) and independent from each other: X1, X2, and X3. These three variables acted as predictor variables or input variables to the neural network. In order to generate the Y output variable of the neural network as a function of the input variables, the following equation was used:

$$Y = 0.0183 \times \exp(4 \times X2) + \tanh(X3) + \text{Error } N(0,\ 0.1) \quad (1)$$

In this way X1 does not have any contribution in the output variable Y. The function that is established between X2 and the output variable is of the exponential type with a range between 0 and 1 for the output variable, while the function between X3 and the output variable is the hyperbolic tangent with a range between –1 and 1 for the output variable. Finally a random error was added with normal distribution, 0 mean and a standard deviation of 0.1.

Given this configuration, an adequate interpretative method should establish that the variable X3 is the one with the greatest effect, followed by the variable X2 with a light effect and the X1 variable with a null effect.

With the aim of correctly training and validating the neural network , the data matrix was divided randomly into three groups: 300 patterns acted as a training set, 100 patterns acted as a validation set, and 100 patterns acted as a test set. The neural network simulated was composed of three input neurons that represented the three predictor variables (X1, X2 and X3), two hidden neurons and an output neuron that represented the function Y. The connection and threshold weights were inicialized repeatedly with different seed values. The neural network that had the greatest yield in the presence of the validation set was the one selected to go on to the test phase, which showed a good fitting in the presence of test data with a RMS error = 0.1112.

Table 1 shows the results obtained from applying the interpretative methods to the selected neural network. For the numeric and analytic sensitivity analysis, the value of the arithmetic mean and the standard deviation of the slope between each input and

output is given, calculated on the training set. Thus, the larger the absolute value of the arithmetic mean, the larger the slope or effect of the input on the network output. Garson's method reveals the percentage of relative importance that each input has on the output obtained by the magnitude of connection weights. It can be seen that the three methods coincide in establishing the correct hierarchy regarding the degree of importance of the input variables. Although the analytic sensitivity analysis gives an estimation of the slope that is more precise in comparison to the numeric sensitivity analysis, this latter is especially useful when discrete variables are included in the model due to the fact that the analytic sensitivity analysis is based on the supposition that all implied variables are of a continuous nature (Sarle, 2000).

Table 1. Results obtained by the interpretative methods in the neural network.

	Numeric sensitivity analysis		**Analitic Sensitivity analysis**		**Garson's method**
	M	SD	M	SD	
X1	0.0352	2.6570	0.0044	0.0010	1.39%
X2	0.1433	2.3980	0.1853	0.0375	23.27%
X3	0.9774	0.9372	1.2261	0.5190	75.34%

By way of example, figure 2 shows the graphic representation of the function between the input X3 and the output learned by the neural network which is correctly adjusted to the hyperbolic tangent sigmoidal function generated by simulation between both variables. This set of results shows the utility of the interpretative methods implemented in the Sensitivity Neural Network in the analysis of the importance of the input variables on the output of a multilayer perceptron network.

The program sensitivity Neural Network works under the operating system Windows, contains a complete Help file, and is available by contacting the authors via e-mail (alfonso.palmer@uib.es).

References

Bishop, C.M. (1995). *Neural networks for pattern recognition.* Oxford: Oxford University Press.

Garson, G.D. (1991). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Sarle, W.S. (2000). *How to measure importance of inputs?* Retrieved November 22, 2001, from ftp://ftp.sas.com/pub/neural/importance.html.

3. Resumen de Resultados y Conclusiones

3.1. Resultados.

El conjunto de resultados obtenidos en los distintos trabajos que componen la tesis, se puede dividir en función de la línea de investigación llevada a cabo.

En la aplicación de las RNA al campo de las conductas adictivas, se creó una red neuronal para la clasificación de sujetos consumidores y no consumidores de éxtasis a partir de 25 items agrupados en cinco categorías temáticas siguiendo los principios de la *Squashing Theory* (Buscema, 1995). Un análisis de sensibilidad sobre el modelo de red creado, permitió identificar los factores de riesgo asociados al consumo de éxtasis.

La red neuronal resultante obtiene un valor AUC = 0.99 (Área bajo la curva ROC, Swets, 1973, 1988) con un SE = 0.005, a partir del grupo de test. Esto significa que la probabilidad en términos de porcentaje de clasificar correctamente un par de sujetos –uno consumidor y otro no consumidor--, seleccionados al azar es del 99%. La creación de cinco submodelos de red, cada uno entrenado a partir de las variables que formaban una categoría temática (Demografía, padres y religión, Ocio, Consumo, Opinión sobre el éxtasis y Personalidad) permitió estudiar el valor predictivo de cada una de las categorías. Así, el valor AUC indica que las dos categorías con mayor poder predictivo son las de ocio (AUC = 0.96 con SE = 0.02) y consumo (AUC = 0.95 con SE = 0.02). La categoría de personalidad alcanza un valor predictivo muy satisfactorio (AUC = 0.88 con SE = 0.04). Por último, las categorías de demografía, padres y religión (AUC = 0.80 con SE = 0.05), y opinión sobre el éxtasis (AUC = 0.74 con SE = 0.06) son las que presentan menor poder predictivo, aunque ambas superan el valor 0.70, límite establecido por Swets (1988) para determinar la utilidad diagnóstica de un modelo.

La aplicación del análisis de sensibilidad sobre el modelo general inicialmente entrenado, puso de manifiesto que las variables con mayor influencia en el consumo de éxtasis son: la cantidad de amigos/as que consumen éxtasis, el consumo de tabaco, la frecuencia en asistir a afters, el estatus económico, el tipo de música preferida y la frecuencia en asistir a fiestas raves. Estos resultados concuerdan con los obtenidos al evaluar el rendimiento de los diferentes submodelos, es decir, las variables de ocio y consumo son las que, en general, tienen mayor influencia sobre el consumo de éxtasis.

En la aplicación de las RNA al análisis de supervivencia, el estudio comparativo entre modelos de red jerárquicos, modelos de red secuenciales y modelo de Cox se realizó mediante la utilización de medidas de resolución (AUC) y calibración (prueba de Hosmer-Lemeshow, Hosmer y Lemeshow, 1980).

La comparación en cuanto a resolución entre el modelo de redes jerárquicas y el modelo de Cox, pone de manifiesto mediante la prueba de Hanley y McNeil (1983) –para la comparación de valores AUC--, que el modelo de red discrimina mejor entre sujetos con cambio y sin cambio de estado respecto al modelo de Cox en todos los intervalos de tiempo estudiados. En relación a la calibración, no se aprecian diferencias entre ambos modelos en cuanto a la bondad de ajuste medida con la prueba de Hosmer-Lemeshow.

La comparación tanto en resolución como en calibración entre el modelo de red jerárquica y el modelo de red secuencial pone de manifiesto que no hay diferencias entre ambos modelos. Más bien, se ha podido observar que el rendimiento en resolución de las redes secuenciales es inferior en numerosos casos.

Finalmente, se ha podido comprobar que ambos modelos de red analizados proporcionan curvas de supervivencia más ajustadas a la realidad en comparación al modelo de Cox, a partir de un ejemplo representativo perteneciente al grupo de test.

Para la realización del estudio comparativo entre los métodos interpretativos dirigidos al análisis del efecto o importancia de las variables de entrada en una red MLP, se diseñó el programa *Sensitivity Neural Network 1.0*. Este programa permite la simulación del comportamiento de una red MLP asociada al algoritmo de aprendizaje *backpropagation*. Un interfaz amigable facilita la manipulación de los aspectos más relevantes del entrenamiento y validación de la red neuronal: selección y visualización de las matrices de datos utilizadas, configuración de la arquitectura y de los parámetros de aprendizaje, inicialización o importación de los pesos, y criterios de parada del entrenamiento. Una vez obtenido el modelo de red, un informe de resultados proporciona información sobre la arquitectura, el valor de los pesos, los valores estimados para cada patrón y la evaluación del rendimiento global del modelo. Como novedad *Sensitivity Neural Network 1.0* implementa un conjunto de métodos que han demostrado en la literatura de RNA ser de utilidad en el análisis del efecto o importancia de las variables de entrada sobre la salida de la red. Los procedimientos

implementados son: el método de Garson (1991) como representativo del análisis basado en la magnitud de los pesos, el método ASA (*Analytic Sensitivity Analysis*) basado en el cálculo de la matriz Jacobiana (Bishop, 1995) y el método NSA (*Numeric Sensitivity Analysis*) (Montaño, Palmer y Fernández, 2002), propuesto por nosotros, que intenta superar las limitaciones de los anteriores métodos mediante el cálculo numérico de la pendiente entre entradas y salidas. Por último, mediante la aplicación del método NSA, se proporciona la representación gráfica de la función subyacente que la red ha aprendido entre cada variable de entrada y la salida.

La comparación entre los métodos interpretativos se realizó en dos estudios paralelos: estudio 1 (Montaño, Palmer y Fernández, 2002) y estudio 2 (Montaño y Palmer, en revisión). En cada estudio se han obtenido mediante simulación cuatro matrices de datos, en cada una de las cuales se manipuló el grado de relación y la naturaleza de las variables de entrada: variables cuantitativas, variables binarias, variables politómicas y, por último, variables cuantitativas y discretas (binarias y politómicas).

Los resultados de la comparación ponen de manifiesto que los diferentes métodos se comportan de forma correcta cuando las variables de entrada son cuantitativas, y cuando son cuantitativas y discretas, a excepción del método basado en el análisis del incremento del error que para esta última configuración de variables, no establece de forma correcta la jerarquía de importancia entre las variables de entrada. Por su parte, el método de Garson, en alguna ocasión, sobrevalora la importancia de variables de entrada que son irrelevantes en la salida de la red. Finalmente, el método NSA y el método ASA obtienen valores promedio muy similares e identifican de forma correcta aquellos casos en los que la relación entre entrada y salida es negativa.

Cuando las variables son binarias, se puede observar que el método NSA es el que representa con más exactitud la realidad, debido a que el índice que proporciona coincide con los valores reales del índice de asociación Phi entre las variables de entrada y la salida de la red. Por otra parte, se puede observar que el método ASA, a pesar de ser el que proporciona valores promedio más próximos al método NSA, presenta una clara discrepancia respecto al valor real. Por último, el método de Garson y el método *weight product* (Tchaban, Taylor y Griffin, 1998) sobrevaloran la importancia de variables de entrada irrelevantes.

Cuando las variables son politómicas, se puede comprobar que el método NSA también es el que mejor se ajusta a la realidad, debido a que es el método que más se aproxima a los valores reales del índice de asociación V de Cramer. Por su parte, el método ASA establece de forma correcta la jerarquía de importancia entre variables, aunque los valores proporcionados por este método no se corresponden con el índice V. Al igual que en el caso anterior, el método de Garson y el método *weight product* sobrevaloran la importancia de variables de entrada que son irrelevantes. Por su parte, el método *weight product* no establece de forma correcta la jerarquía de importancia entre las variables de entrada.

Por último, la obtención de la representación gráfica del método NSA mediante el programa *Sensitivity Neural Network 1.0*, ha permitido comprobar que efectivamente la red neuronal aprende correctamente las funciones subyacentes entre las variables de entrada y la salida, reproduciendo con bastante fidelidad la forma de las funciones establecidas *a priori* en la simulación.

3.2. Discusión y conclusiones finales.

En esta tesis se han descrito tres líneas de investigación en torno a la utilización de RNA en el ámbito del análisis de datos. Cada una de ellas ha operado en un nivel de análisis diferente. Por un lado, se han utilizado las RNA como modelos de predicción en el campo sustantivo de las conductas adictivas, explotando las diferentes posibilidades ofrecidas por las RNA en la vertiente aplicada. Por otro lado, se han aplicado las RNA en un campo propio de la metodología y la estadística, esto es, el análisis de supervivencia, con el objeto de realizar un estudio comparativo respecto a los modelos estadísticos clásicos. Por último, centrando la atención en las propias RNA, se ha intentado superar el principal inconveniente que presenta esta tecnología desde un punto de vista estadístico, a saber: la dificultad en estudiar el efecto o importancia de las variables de entrada en una red MLP. Un denominador común a estas tres líneas de investigación consiste en que todas ellas constituyen aspectos relevantes en el análisis de datos y que a pesar de ello, son líneas de investigación minoritarias a juzgar por los trabajos realizados hasta la fecha. Nuestra labor trata de dar respuesta a diversos interrogantes planteados en cada una de estas líneas de investigación. A continuación, se presenta una discusión global de los resultados obtenidos.

En la aplicación de RNA al campo de las conductas adictivas se ha pretendido, por un lado, construir una red neuronal capaz de predecir el consumo de éxtasis en la población de jóvenes europeos y, por otro lado, identificar los factores de riesgo asociados al consumo de esta sustancia mediante la aplicación de un análisis de sensibilidad.

El Centro de Investigación Semeion dirigido por Buscema es pionero en el uso de RNA en el ámbito de las adicciones, habiéndose centrado de forma exclusiva en la utilización de muestras clínicas. Los investigadores de dicho centro han construido diferentes modelos de red dirigidos a la discriminación de sujetos adictos –principalmente a la heroína y al alcohol— respecto a sujetos no consumidores y a la extracción de las características prototípicas de los sujetos adictos.

La perspectiva adoptada en nuestro trabajo difiere en una serie de aspectos en relación a la línea de investigación del equipo de Buscema. Por un lado, la muestra utilizada pertenece a la población general de jóvenes europeos. En este caso, no se ha pretendido identificar sujetos adictos, sino sujetos que habitualmente consumen éxtasis, con el fin de realizar una labor preventiva. La tarea de discriminación ente consumidores y no consumidores por parte de la red neuronal es más compleja *a priori* que la discriminación entre adictos y no adictos. Esto es debido a que las diferencias en cuanto a características bio-psico-sociales que pueden actuar como variables predictoras son más difusas en el primer caso que en el segundo. Por otra parte, se ha querido comprobar, en contra de la concepción tradicional, que los pesos y valores de activación de una red neuronal pueden dar información acerca del grado de influencia de las variables de entrada sobre la salida de la red.

La red neuronal construida ha sido capaz de predecir el consumo de éxtasis en la población de jóvenes europeos a partir de las respuestas dadas a un cuestionario, con un nivel de error del 1%. Este resultado mejora los obtenidos por el equipo de Buscema. Así, Buscema, Intraligi y Bricolo (1998) desarrollaron varios modelos de red neuronal para la predicción de la adicción a la heroína con un nivel de eficacia siempre superior al 91%. Por su parte, Maurelli y Di Giulio (1998) obtuvieron un modelo de red capaz de predecir el grado de alcoholismo de un sujeto, a partir de los resultados de varios test biomédicos, con una capacidad de predicción del 93%.

El análisis de sensibilidad realizado ha mostrado la importancia de aspectos del individuo no directamente relacionados con el consumo de éxtasis como el hábito de

consumo de alcohol y tabaco, preferencias respecto a los lugares de ocio, y características de personalidad como el grado desinhibición o de desviación social.

Este conjunto de resultados tienen una serie de implicaciones tanto a nivel teórico para la problemática de las conductas adictivas como a nivel metodológico para el análisis de datos.

Desde un punto de vista práctico, la ONG IREFREA ha utilizado la red neuronal construida por nuestro equipo para la clasificación de sujetos que han entrado a formar parte recientemente en la base de datos de la organización. El rendimiento de la red neuronal puede calificarse de excelente. Por otra parte, la identificación de los factores de riesgo asociados al consumo de éxtasis mediante el análisis de sensibilidad, ha motivado la confección de un nuevo cuestionario de preguntas que profundizan en los aspectos que han demostrado ser relevantes en la predicción de esta conducta. Finalmente, se han elaborado las directrices generales para el desarrollo de un plan de prevención e intervención que permita actuar sobre las variables de riesgo identificadas.

Desde un punto de vista metodológico, se ha comprobado que las RNA resultan de suma utilidad en el estudio de los fenómenos de comportamiento tanto individuales como sociales, los cuales estan determinados en la mayoría de casos por multitud de factores conocidos y desconocidos, pudiéndose establecer interacciones complejas entre las variables implicadas. Por otra parte, las RNA no sólo han sido utilizadas como herramientas de predicción sino también de explicación, pudiéndose cuantificar la contribución de cada variable de entrada sobre el valor predicho por la red neuronal.

En la aplicación de RNA al análisis de supervivencia, se han utilizado dos modelos exclusivamente neuronales –redes jerárquicas y redes secuenciales— con capacidad para el manejo de datos de supervivencia. Como se ha comentado en la introducción, un modelo estadístico o red neuronal convencional no posee tal capacidad debido a la presencia de datos censurados y la posible introducción de variables dependientes del tiempo.

En ambos modelos de red utilizados, la información parcial proporcionada por los datos censurados es utilizada en aquellos submodelos de red para los que se tiene información del cambio de estado en el intervalo de tiempo correspondiente. Por ejemplo, los datos de un sujeto al que se le haya realizado el seguimiento hasta el tercer intervalo

considerado, serán usados en los submodelos correspondientes al primer, segundo y tercer intervalo, pero no en los submodelos correspondientes a los siguientes intervalos de tiempo. Si bien, en nuestra investigación no se han utilizado variables dependientes del tiempo, éstas también se pueden incorporar fácilmente debido a que cada submodelo puede recibir, en cada momento temporal, un valor diferente respecto a las variables explicativas para un mismo sujeto.

Los conjuntos de datos utilizados en las diferentes investigaciones donde se aplica RNA al análisis de supervivencia, provienen, de forma prácticamente exclusiva, del campo de la medicina. Concretamente, el equipo de Ohno-Machado ha aplicado las redes jerárquicas y secuenciales al análisis de supervivencia en sujetos con enfermedad coronaria y en sujetos con SIDA (Ohno-Machado, 1996; Ohno-Machado y Musen, 1997a; Ohno-Machado y Musen, 1997b). Con nuestra investigación hemos tratado de averiguar en qué medida los resultados obtenidos en los trabajos de Ohno-Machado se pueden extrapolar al área de las Ciencias del Comportamiento, utilizando una matriz de datos perteneciente al campo de las conductas adictivas.

La comparación llevada a cabo en cuanto a poder predictivo entre los modelos presentados ha permitido responder a las hipótesis planteadas en la investigación, como vamos a ver a continuación.

En primer lugar, el modelo de redes jerárquicas presenta un rendimiento superior en cuanto a resolución frente al modelo de Cox, mientras que ambos modelos han mostrado una calibración similar. Este resultado coincide con los obtenidos en trabajos de simulación (Pitarque, Roy y Ruíz, 1998) donde se observa que las RNA tienen un rendimiento superior en cuanto a capacidad de discriminación o clasificación, pero no suponen una mejora en cuanto a bondad de ajuste respecto a los modelos estadísticos clásicos. Por su parte, los resultados obtenidos por Ohno-Machado (1996) mediante la comparación entre redes jerárquicas y modelo de Cox muestran una resolución mejor por parte de las RNA, aunque esta superioridad no es significativa en la mitad de los intervalos de tiempo estudiados. La comparación entre ambos modelos en cuanto a calibración, muestra un pobre rendimiento en la mayoría de intervalos de tiempo por parte del modelo de Cox respecto al modelo de red, a diferencia de los resultados obtenidos en nuestra investigación.

En segundo lugar, el modelo de redes secuenciales no supone una mejora en rendimiento respecto al modelo de redes jerárquicas. En cuanto a resolución, se ha podido observar que incluso el rendimiento de las redes secuenciales es inferior en numerosos casos. En cuanto a calibración, se aprecia que en general el rendimiento mejora en las redes secuenciales cuando se utiliza como intervalo informativo el intervalo inmediatamente posterior. Sin embargo, dado que el rendimiento de las redes jerárquicas en calibración es bueno, la mejora observada en las redes secuenciales no es relevante. La conclusión a la que podemos llegar, por tanto, es que la inclusión de información acerca de la función de supervivencia estimada en un intervalo de tiempo dado, no mejora la predicción de la función de supervivencia en otro intervalo de tiempo. Por su parte, los resultados obtenidos por Ohno-Machado (1996) mediante la comparación entre redes jerárquicas y redes secuenciales muestran que en algunas ocasiones la resolución de las redes secuenciales es superior, sin embargo, no se observan diferencias entre ambos modelos en cuanto a calibración, al igual que los resultados obtenidos en nuestra investigación.

En tercer lugar, los modelos de red proporcionan curvas de supervivencia más ajustadas a la realidad que el modelo de Cox. Así, se ha podido observar que las funciones de supevivencia de las RNA experimentan un claro decremento en el intervalo temporal en el que el sujeto realiza el cambio de estado, a diferencia del modelo de Cox que presenta una disminución más conservadora en la función de supervivencia.

El estudio comparativo realizado, ha permitido observar una serie de características diferenciales entre los modelos de red y el modelo de Cox, que pueden explicar las diferencias halladas en cuanto a rendimiento entre ambas perspectivas.

En primer lugar, las RNA no están supeditadas al cumplimiento de condiciones estadísticas tales como el supuesto de proporcionalidad. En los datos de supervivencia utilizados se ha comprobado que una de las variables explicativas, Durac (duración de estancia en tratamiento), no cumple el supuesto de proporcionalidad. Como procedimiento habitual, la variable Durac fue excluida del modelo de Cox para ser utilizada como variable de estratificación con dos estratos. Esta medida supone una pérdida de información considerable, dado que a pesar de ser utilizada la variable Durac como variable de estrato, el modelo resultante no tiene en cuenta el valor cuantitativo que toma el sujeto en esa variable a la hora de realizar predicciones sobre la función de

supervivencia. En contraposición, los modelos de red utilizan toda la información disponible en las variables explicativas sin introducir restricciones sobre la estructura de los datos.

En segundo lugar, en el modelo de Cox, como en cualquier modelo de regresión clásico, se deben explicitar tanto las posibles interacciones entre variables predictoras como las funciones complejas que se puedan establecer entre variables predictoras y valor predicho. Como consecuencia, los modelos resultantes pueden llegar a ser sumamente complejos partiendo de un número reducido de variables. De hecho, en la mayoría de aplicaciones no se suelen introducir términos de interacción (Ohno-Machado, 1996). Así, en nuestra investigación, únicamente se pudo introducir en el modelo de Cox los términos de interacción de primer orden, ya que la introducción de términos de orden superior no permitía al método de estimación alcanzar la convergencia de los parámetros del modelo. En los modelos de red neuronal no es necesario introducir de forma explícita términos de interacción entre predictores ni funciones concretas entre predictores y variable de respuesta, debido a que son aprendidos de forma automática en el proceso de entrenamiento del modelo. De esta forma, las RNA pueden resolver problemas de alta dimensionalidad mediante la utilización de arquitecturas relativamente sencillas en relación al modelo de Cox.

Como es obvio, tampoco debemos pensar que las RNA son la panacea que permite solucionar cualquier problema, ya que los modelos de red analizados también cuentan con algunos inconvenientes. Las redes jerárquicas y secuenciales no tratan la variable tiempo de supervivencia como una variable continua al igual que el modelo de Cox, sino que ésta debe ser transformada en intervalos discretos de tiempo. Esta estrategia no permite optimizar la información proporcionada por los datos. Por ejemplo, dos sujetos que realizan el cambio, uno al principio y otro al final de un mismo intervalo, son tratados de igual forma. Por tanto, desde este punto de vista, el análisis de supervivencia con RNA se reduce a un problema de clasificación en cada intervalo de tiempo definido: 0 = cambio, 1 = no cambio. También se abren interrogantes de difícil respuesta acerca del número de intervalos de tiempo y puntos de corte más adecuados. Por otra parte, el uso de estos modelos de red supone una labor de preprocesamiento de datos considerable. Para cada intervalo de tiempo, se debe especificar el estado del sujeto; en caso de ser un dato censurado, éste debe ser eliminado para el intervalo actual y los posteriores.

A modo de conclusión, se puede decir a la luz de los resultados obtenidos que las RNA suponen una alternativa a los modelos de supervivencia tradicionales para aquellos casos en que la relación entre las variables sea de elevada complejidad, el número de registros sea suficientemente grande, el número de variables implicadas sea alto y con posibles interacciones desconocidas y, finalmente, no se puedan asumir los supuestos del modelo estadístico.

En el estudio del efecto de las variables de entrada en una red MLP se planteó el diseño del programa *Sensitivity Neural Network 1.0* y la realización de un estudio comparativo sobre el rendimiento de los métodos interpretativos implementados en el programa.

El desarrollo de *Sensitivity Neural Network 1.0* ha permitido cubrir una importante laguna en el campo de la simulación de RNA mediante *software*. La mayoría de programas simuladores se han centrado en el entrenamiento y validación del modelo de red con fines predictivos. Sin embargo, apenas se ha prestado atención al análisis de la influencia de las variables de entrada sobre la salida de la red, asumiendo erróneamente que las RNA no son susceptibles de tal análisis bajo su condición de "caja negra". La novedad de *Sensitivity Neural Network 1.0* no reside, por tanto, en la simulación que realiza del aprendizaje (mediante el algoritmo *backpropagation*) y funcionamiento de una red MLP, sino en la incorporación de un conjunto de métodos numéricos dirigidos al estudio del efecto de las entradas sobre las salidas. Por otra parte, hace uso de un sencillo formato de lectura de pesos que permite la importación de los parámetros obtenidos con otro programa simulador de RNA. Esto es especialmente útil en aquellos casos en los que se ha obtenido una configuración de pesos mediante un algoritmo diferente al *backpropagation* y estamos interesados en estudiar la influencia de las variables de entrada.

En la introducción de la tesis se ha realizado una descripción de los diferentes procedimientos propuestos a lo largo de la década de los 90 para el estudio del efecto de las variables de entrada en una red MLP. Sin embargo, también se ha apuntado que apenas existen trabajos orientados a la validación de estos métodos, los cuales presentan limitaciones importantes. En este sentido, cabe destacar el trabajo de Sarle (2000) en el que se analiza a partir de una sencilla matriz de datos el comportamiento de varios métodos como el análisis basado en la magnitud de los pesos o el cálculo de la matriz Jacobiana. Los resultados ponen de manifiesto que ninguno de los métodos analizados

permiten estudiar de forma correcta el efecto o importancia de las variables de entrada de la red.

El método NSA (*Numeric Sensitivity Analysis*) presentado por nuestro equipo, puede considerarse como una versión mejorada del análisis de sensibilidad aplicado en el trabajo *Predicción del consumo de éxtasis a partir de redes neuronales artificiales* (Palmer, Montaño y Calafat, 2000) y fue diseñado con el objeto de superar las limitaciones presentadas por los anteriores métodos.

El estudio realizado por nuestro equipo puede considerarse como el primer trabajo comparativo realizado de forma sistemática sobre un conjunto de métodos interpretativos aplicados a una red MLP. Los resultados de la comparación ponen de manifiesto que el método NSA, es el que, de forma global, permite evaluar con mayor exactitud la importancia o efecto de las variables de entrada con independencia de su naturaleza (cuantitativa o discreta), superando a los demás métodos.

Más concretamente, se ha podido observar que el método NSA y el método ASA (*Analytic Sensitivity Analysis*) proporcionan valores promedio similares cuando las variables implicadas son cuantitativas. El método ASA es perfectamente válido en estos casos, teniendo en cuenta además la baja variabilidad que proporciona. En este sentido, trabajos anteriores corroboran esta conclusión (Harrison, Marshall y Kennedy, 1991; Takenaga, Abe, Takatoo, Kayama, Kitamura y Okuyama, 1991; Guo y Uhrig, 1992; Castellanos, Pazos, Ríos y Zafra, 1994; Engelbrecht, Cloete y Zurada, 1995; Bahbah y Girgis, 1999; Rambhia, Glenny y Hwang, 1999).

Sin embargo, cuando las variables implicadas son discretas (binarias y politómicas), el método NSA es el más adecuado debido a que los valores que proporciona coinciden con el índice Phi en el caso de variables binarias y se aproximan considerablemente al índice V de Cramer en el caso de variables politómicas. El método ASA es el que proporciona valores más próximos al método NSA cuando las variables son binarias, demostrando ser robusto a pesar de que las variables implicadas no sean de naturaleza continua, condición enunciada por Sarle (2000) para poder aplicar este método asumiendo que, de lo contrario, éste no proporciona información significativa acerca de la importancia de las variables de entrada.

Cuando las variables son binarias o politómicas, el método de Garson y el método *weight product* sobrevaloran la importancia de variables que son irrelevantes para la salida de la red. El método *weight product*, cuando las variables son politómicas, y el método basado en el cálculo del incremento en la función RMC error, cuando las variables son cuantitativas y discretas, no son capaces de establecer correctamente la jerarquía de importancia entre las variables de entrada. Trabajos como el de Gedeon (1997), Tchaban, Taylor y Griffin (1998) y Sarle (2000) coinciden en apuntar la baja fiabilidad de estos métodos.

Por otra parte, con el método NSA la interpretación del efecto de una variable es más sencilla ya que el índice que proporciona está acotado en el intervalo [-1, 1] a diferencia de los valores que teóricamente pueden proporcionar, por ejemplo, el método ASA y el método *weight product*.

Por último, el método NSA incorpora un procedimiento que permite representar gráficamente la función aprendida por la red ente una variable de entrada y la salida. Esta representación gráfica aporta información relevante que complementa la información proporcionada por los índices numéricos, debido a que en muchos casos un índice de resumen no es suficiente para reflejar la función subyacente entre variables.

Conclusiones finales.

Una vez presentada en detalle la discusión de los resultados obtenidos en las diferentes líneas de investigación de la tesis, estamos en disposición de enunciar una serie de conclusiones acerca de las contribuciones realizadas en este trabajo.

En primer lugar, las RNA son capaces de predecir el consumo de éxtasis con un margen de error pequeño a partir de las respuestas dadas a un cuestionario. Desde una perspectiva explicativa, el análisis de sensibilidad aplicado al modelo de red ha identificado los factores asociados al consumo de esta sustancia. Esto demuestra que los buenos resultados obtenidos por las RNA en diferentes áreas de conocimiento como la medicina, la ingeniería o la biología, se extienden también al campo de las Ciencias del Comportamiento.

En segundo lugar, los modelos de redes jerárquicas y secuenciales permiten el manejo de datos de supervivencia superando en algunos aspectos el rendimiento del modelo que

tradicionalmente ha sido utilizado hasta el momento, el modelo de Cox. A diferencia de este modelo, las RNA no se ven afectadas por el cumplimiento del supuesto de proporcionalidad, tampoco es necesario introducir de forma explícita términos de interacción entre predictores ni funciones concretas entre predictores y variable de respuesta, debido a que son aprendidos de forma automática en el proceso de entrenamiento del modelo.

Por último, el método NSA propuesto por nosotros es el que permite evaluar con mayor exactitud la importancia o efecto de las variables de entrada de una red MLP. Con ello, se ha pretendido mostrar que una red neuronal no es una "caja negra", sino más bien un instrumento de predicción capaz de proporcionar claves acerca de las variables que están determinando la estimación realizada por el modelo. La creación del programa *Sensitivity Neural Network 1.0* ha supuesto un avance importante en este sentido. Ahora el usuario cuenta con un programa que incorpora un conjunto de procedimientos numéricos y gráficos que han demostrado empíricamente ser de utilidad en el análisis del efecto de las variables de entrada de una RNA.

A pesar de estos importantes logros alcanzados, esperamos que la principal contribución de este trabajo haya sido la de incentivar al lector en la aplicación de nuevas tecnologías, como las redes neuronales, en el ámbito de la estadística y el análisis de datos.

Anexo 1:
Otras Publicaciones

Tutorial sobre redes neuronales artificiales: el Perceptrón multicapa.

REVISTA ELECTRÓNICA DE PSICOLOGÍA
Vol. 5, No. 2, Julio 2001
ISSN 1137-8492

Tutorial sobre Redes Neuronales Artificiales: El Perceptrón Multicapa

Palmer, A., Montaño, J.J. y Jiménez, R.
Área de Metodología de las Ciencias del Comportamiento.
Facultad de Psicología.
Universitat de les Illes Balears.
e-Mail: alfonso.palmer@uib.es

1.- Introducción

Las Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas (Palmer y Montaño, 1999). Consisten en un gran número de elementos simples de procesamiento llamados nodos o neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. En los pesos se encuentra el conocimiento que tiene la RNA acerca de un determinado problema.

En la Web podemos encontrar un sinfín de introducciones al campo de las RNA. En este sentido, el Pacific Northwest National Laboratory ofrece un listado de excelentes introducciones *on line* a este campo.

La utilización de las RNA puede orientarse en dos direcciones, bien como modelos para el estudio del sistema nervioso y los fenómenos cognitivos, bien como herramientas para la resolución de problemas prácticos como la clasificación de patrones y la predicción de funciones. Desde esta segunda perspectiva que será la adoptada en este documento, las RNA han sido aplicadas de forma satisfactoria en la predicción de diversos problemas en diferentes áreas de conocimiento --biología, medicina, economía, ingeniería, psicología, etc. (Arbib, 1995; Simpson, 1995; Arbib, Erdi y Szentagothai, 1997)--; obteniendo excelentes resultados respecto a los modelos derivados de la estadística clásica (De Lillo y Meraviglia, 1998; Jang, 1998; Waller, Kaiser, Illian et al., 1998; Arana, Delicado y Martí-Bonmatí, 1999; Takahashi, Hayasawa y Tomita, 1999). El paralelismo de cálculo, la memoria distribuida y la adaptabilidad al entorno, han convertido a las RNA en potentes instrumentos con capacidad para aprender relaciones entre variables sin necesidad de imponer presupuestos o restricciones de partida en los datos.

Actualmente, existen unos 40 paradigmas de RNA que son usados en diversos campos de aplicación (Sarle, 1998). Entre estos paradigmas, el más ampliamente utilizado es el perceptrón multicapa asociado al algoritmo de aprendizaje *backpropagation error* (propagación del error hacia atrás), también denominado red *backpropagation* (Rumelhart, Hinton y Williams, 1986). La popularidad del perceptrón multicapa se debe principalmente a que es capaz de actuar como un aproximador universal de funciones

(Funahashi, 1989; Hornik, Stinchcombe y White, 1989). Más concretamente, una red *backpropagation* conteniendo al menos una capa oculta con suficientes unidades no lineales puede aprender cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida. Esta propiedad convierte a las redes perceptrón multicapa en herramientas de propósito general, flexibles y no lineales.

En el presente documento, nos proponemos realizar la descripción del funcionamiento de una red perceptrón multicapa entrenada mediante la regla de aprendizaje *backpropagation*. Con el objeto de alcanzar una mejor comprensión, tal descripción irá acompañada de applets y software ilustrativos, los cuales estarán a disposición del lector via internet.

2.- El perceptrón multicapa

Rumelhart, Hinton y Williams (1986) formalizaron un método para que una red del tipo perceptrón multicapa aprendiera la asociación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Este método, conocido como *backpropagation error* (propagación del error hacia atrás) --también denominado método de gradiente decreciente--, ya había sido descrito anteriormente por Werbos (1974), Parker (1982) y Le Cun (1985), aunque fue el *Parallel Distributed Processing Group* (grupo PDP) --Rumelhart y colaboradores--, quien realmente lo popularizó.

La importancia de la red *backpropagation* consiste en su capacidad de organizar una representación interna del conocimiento en las capas ocultas de neuronas, a fin de aprender la relación que existe entre un conjunto de entradas y salidas. Posteriormente, aplica esa misma relación a nuevos vectores de entrada con ruido o incompletos, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje. Esta característica importante es la capacidad de generalización, entendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

2.1.- Arquitectura

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta (Funahashi, 1989; Hornik, Stinchcombe y White, 1989). En la figura 1 podemos observar un perceptrón típico formado por una capa de entrada, una capa oculta y una de salida.

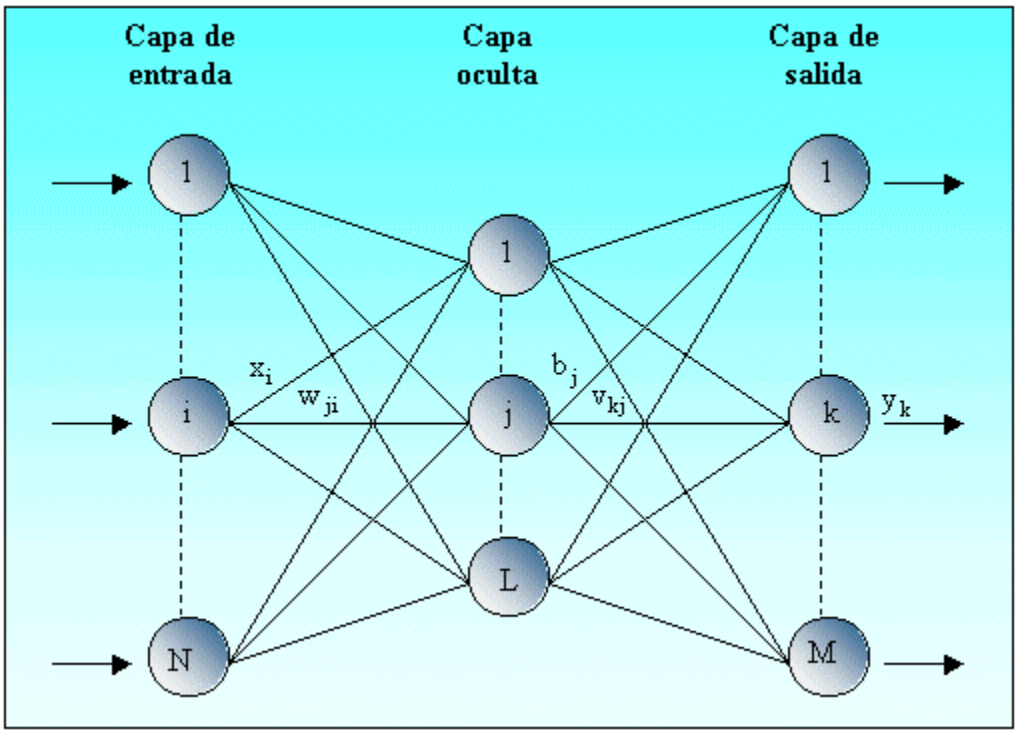


Figura 1: Perceptrón multicapa.

En este tipo de arquitectura, las conexiones entre neuronas son siempre hacia delante, es decir, las conexiones van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa; no hay conexiones laterales --esto es, conexiones entre neuronas pertenecientes a una misma capa--, ni conexiones hacia atrás --esto es, conexiones que van desde una capa hacia la capa anterior. Por tanto, la información siempre se transmite desde la capa de entrada hacia la capa de salida.

En el presente documento, hemos considerado w_{ji} como el peso de conexión entre la neurona de entrada i y la neurona oculta j, y v_{kj} como el peso de conexión entre la neurona oculta j y la neurona de salida k.

2.2.- Algoritmo *backpropagation*

En el algoritmo *backpropagation* podemos considerar, por un lado, una etapa de funcionamiento donde se presenta, ante la red entrenada, un patrón de entrada y éste se transmite a través de las sucesivas capas de neuronas hasta obtener una salida y, por otro lado, una etapa de entrenamiento o aprendizaje donde se modifican los pesos de la red de manera que coincida la salida deseada por el usuario con la salida obtenida por la red ante la presentación de un determinado patrón de entrada.

2.2.1.- Etapa de funcionamiento

Cuando se presenta un patrón p de entrada X_p: $x_{p1},...,x_{pi},...,x_{pN}$, éste se transmite a través de los pesos w_{ji} desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos v_{kj} hacia la capa de salida, donde aplicando la misma operación que en el caso anterior, las neuronas de esta última capa proporcionan la salida de la red. Este proceso se puede explicar matemáticamente de la siguiente manera:

La entrada total o neta que recibe una neurona oculta j, net_{pj}, es:

$$net_{pj} = \sum_{i=1}^{N} w_{ji} x_{pi} + \theta_j$$

donde θ es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida igual a 1.

El valor de salida de la neurona oculta j, b_{pj}, se obtiene aplicando una función f(.) sobre su entrada neta:

$$b_{pj} = f(net_{pj})$$

De igual forma, la entrada neta que recibe una neurona de salida k, net_{pk}, es:

$$net_{pk} = \sum_{j=1}^{L} v_{kj} b_{pj} + \theta_k$$

Por último, el valor de salida de la neurona de salida k, y_{pk}, es:

$$y_{pk} = f(net_{pk})$$

2.2.2.- Etapa de aprendizaje

En la etapa de aprendizaje, el objetivo que se persigue es hacer mínima la discrepancia o error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de patrones denominado grupo de entrenamiento. Por este motivo, se dice que el aprendizaje en las redes *backpropagation* es de tipo supervisado, debido a el usuario (o supervisor) determina la salida deseada ante la presentación de un determinado patrón de entrada.

La función de error que se pretende minimizar para cada patrón p viene dada por:

$$E_p = \frac{1}{2} \sum_{k=1}^{M} (d_{pk} - y_{pk})^2$$

donde d_{pk} es la salida deseada para la neurona de salida k ante la presentación del patrón p. A partir de esta expresión se puede obtener una medida general de error mediante:

$$E = \sum_{p=1}^{P} E_p$$

La base matemática del algoritmo *backpropagation* para la modificación de los pesos es la técnica conocida como gradiente decreciente (Rumelhart, Hinton y Williams, 1986). Teniendo en cuenta que E_p es función de todos los pesos de la red, el gradiente de E_p es un vector igual a la derivada parcial de E_p respecto a cada uno de los pesos. El gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta --es decir, la dirección negativa--, determina el decremento más

rápido en el error. Por tanto, el error puede reducirse ajustando cada peso en la dirección:

$$-\sum_{p=1}^{P}\frac{\partial E_p}{\partial w_{ji}}$$

Vamos a ilustrar el proceso de aprendizaje de forma gráfica: El conjunto de pesos que forma una red neuronal puede ser representado por un espacio compuesto por tantas dimensiones como pesos tengamos. Supongamos para simplificar el problema que tenemos una red formada por dos pesos, el paisaje se puede visualizar como un espacio de dos dimensiones. Por otra parte, hemos comentado que el error cometido es función de los pesos de la red; de forma que en nuestro caso, a cualquier combinación de valores de los dos pesos, le corresponderá un valor de error para el conjunto de entrenamiento. Estos valores de error se pueden visualizar como una superficie, que denominaremos superficie del error.

Como se muestra en la figura 2(A), la superficie del error puede tener una topografía arbitrariamente compleja.

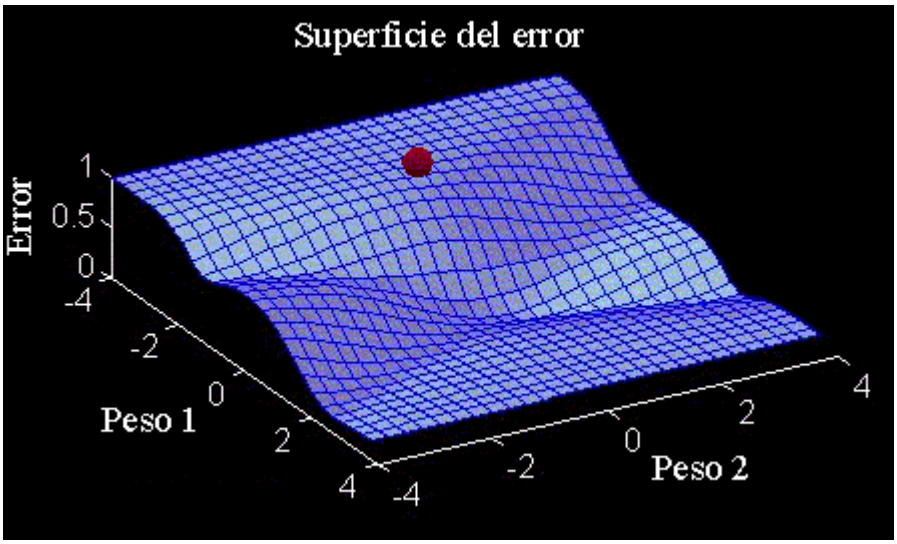


Figura 2 (A): Superficie del error.

Con la imagen de la superficie del error en mente, el proceso de entrenamiento comienza en un determinado punto, representado por la bola roja, definido por los pesos

iniciales de la red (figura 2(A)). El algoritmo de aprendizaje se basa en obtener información local de la pendiente de la superficie --esto es, del gradiente--, y a partir de esa información modificar iterativamente los pesos de forma proporcional a dicha pendiente, a fin de asegurar el descenso por la superficie del error hasta alcanzar el mínimo más cercano desde el punto de partida. La figura 2(B) muestra el proceso descrito mediante la representación del descenso de la bola roja hasta alcanzar una llanura.

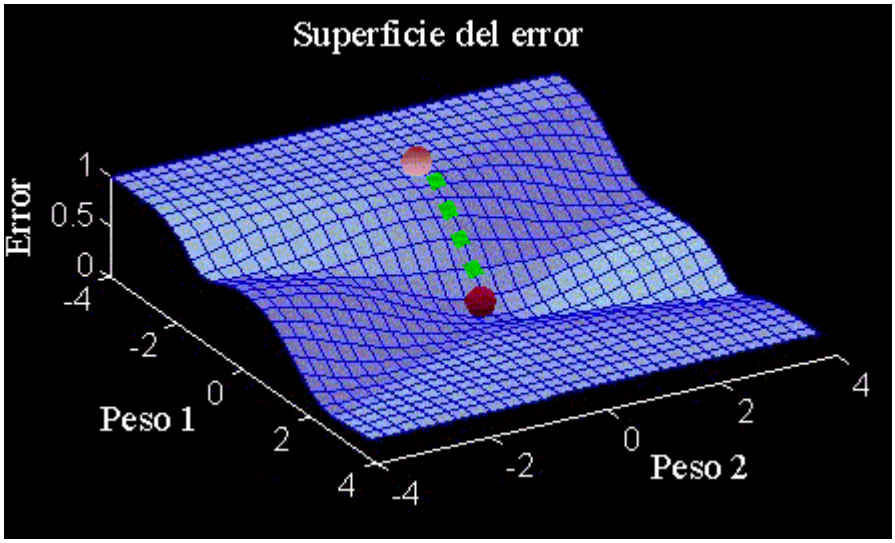


Figura 2 (B): Superficie del error.

Con un número mayor de pesos el espacio se convierte en un plano multidimensional inimaginable, aunque se seguirían aplicando los mismos principios comentados en el ejemplo.

Un peligro que puede surgir al utilizar el método de gradiente decreciente es que el aprendizaje converja en un punto bajo, sin ser el punto más bajo de la superficie del error. Tales puntos se denominan mínimos locales para distinguirlos del punto más bajo de esta superficie, denominado mínimo global. Sin embargo, el problema potencial de los mínimos locales se dan en raras ocasiones en datos reales (Rzempoluck, 1998).

A nivel práctico, la forma de modificar los pesos de forma iterativa consiste en aplicar la regla de la cadena a la expresión del gradiente y añadir una tasa de aprendizaje η. Así, cuando se trata del peso de una neurona de salida:

$$\Delta v_{kj}(n+1) = \eta \sum_{p=1}^{P} \delta_{pk} b_{pj}$$

donde

$$\delta_{pk} = (d_{pk} - y_{pk}) f'(net_{pk})$$

y n indica la iteración. Cuando se trata del peso de una neurona oculta:

$$\Delta w_{ji}(n+1) = \eta \sum_{p=1}^{P} \delta_{pj} x_{pi}$$

donde

$$\delta_{pj} = f'(net_{pj}) \sum_{k=1}^{M} \delta_{pk} v_{kj}$$

Se puede observar que el error o valor delta asociado a una neurona oculta j, viene determinado por la suma de los errores que se cometen en las k neuronas de salida que reciben como entrada la salida de esa neurona oculta j. De ahí que el algoritmo también se denomine propagación del error hacia atrás.

Para la modificación de los pesos, la actualización se realiza después de haber presentado todos los patrones de entrenamiento. Este es el modo habitual de proceder y se denomina aprendizaje por lotes o modo *batch*. Existe otra modalidad denominada aprendizaje en serie o modo *on line* consistente en actualizar los pesos tras la presentación de cada patrón de entrenamiento. En este modo, se debe tener presente que el orden en la presentación de los patrones debe ser aleatorio, puesto que si siempre se siguiese un mismo orden, el entrenamiento estaría viciado a favor del último patrón del conjunto de entrenamiento, cuya actualización, por ser la última, siempre predominaría sobre las anteriores (Martín del Brío y Sanz, 1997).

Con el fin de acelerar el proceso de convergencia de los pesos, Rumelhart, Hinton y Williams (1986) sugirieron añadir en la expresión del incremento de los pesos un factor

momento, α, el cual tiene en cuenta la dirección del incremento tomada en la iteración anterior. Así, cuando se trata del peso de una neurona de salida:

$$\Delta v_{kj}(n+1) = \eta\left(\sum_{p=1}^{P} \delta_{pk} b_{pj}\right) + \alpha \Delta v_{kj}(n)$$

Cuando se trata del peso de una neurona oculta:

$$\Delta w_{ji}(n+1) = \eta\left(\sum_{p=1}^{P} \delta_{pj} x_{pi}\right) + \alpha \Delta w_{ji}(n)$$

En el apartado 3.3.3. se explica con más detalle el papel que juega la tasa de aprendizaje y el factor momento en el proceso de aprendizaje.

2.3.- Variantes del algoritmo *backpropagation*

Desde que en 1986 se presentara la regla *backpropagation*, se han desarrollado diferentes variantes del algoritmo original. Estas variantes tienen por objeto acelerar el proceso de aprendizaje. A continuación, comentaremos brevemente los algoritmos más relevantes.

La regla *delta-bar-delta* (Jacobs, 1988) se basa en que cada peso tiene una tasa de aprendizaje propia, y ésta se puede ir modificando a lo largo del entrenamiento. Por su parte, el algoritmo QUICKPROP (Fahlman, 1988) modifica los pesos en función del valor del gradiente actual y del gradiente pasado. El algoritmo de gradiente conjugado (Battiti, 1992) se basa en el cálculo de la segunda derivada del error con respecto a cada peso, y en obtener el cambio a realizar a partir de este valor y el de la derivada primera. Por último, el algoritmo RPROP (*Resilient propagation*) (Riedmiller y Braun, 1993) es un método de aprendizaje adaptativo parecido a la regla *delta-bar-delta*, donde los pesos se modifican en función del signo del gradiente, no en función de su magnitud.

3.- Fases en la aplicación de un perceptrón multicapa

En el presente apartado se van a exponer los pasos que suelen seguirse en el diseño de una aplicación neuronal (Palmer, Montaño y Calafat, 2000). En general, una red del tipo perceptrón multicapa intentará resolver dos tipos de problemas. Por un lado, los problemas de predicción consisten en la estimación de una variable continua de salida a partir de la presentación de un conjunto de variables predictoras de entrada (discretas y/o continuas). Por otro lado, los problemas de clasificación consisten en la asignación de la categoría de pertenencia de un determinado patrón a partir de un conjunto de variables predictoras de entrada (discretas y/o continuas).

3.1.- Selección de las variables relevantes y preprocesamiento de los datos

Para obtener una aproximación funcional óptima, se deben elegir cuidadosamente las variables a emplear. Más concretamente, de lo que se trata es de incluir en el modelo las variables predictoras que realmente predigan la variable dependiente o de salida, pero que a su vez no covaríen entre sí (Smith, 1993). La introducción de variables irrelevantes o que covaríen entre sí, puede provocar un sobreajuste innecesario en el modelo. Este fenómeno aparece cuando el número de parámetros o pesos de la red resulta excesivo en relación al problema a tratar y al número de patrones de entrenamiento disponibles. La consecuencia más directa del sobreajuste es una disminución sensible en la capacidad de generalización del modelo que como hemos mencionado, representa la capacidad de la red de proporcionar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento.

Un procedimiento útil para la selección de las variables relevantes (Masters, 1993) consiste en entrenar la red con todas las variables de entrada y, a continuación, ir eliminando una variable de entrada cada vez y reentrenar la red. La variable cuya eliminación causa el menor decremento en la ejecución de la red es eliminada. Este procedimiento se repite sucesivamente hasta que llegados a un punto, la eliminación de más variables implica una disminución sensible en la ejecución del modelo.

Una vez seleccionadas las variables que van a formar parte del modelo, se procede al preprocesamiento de los datos para adecuarlos a su tratamiento por la red neuronal.

Cuando se trabaja con un perceptrón multicapa es muy aconsejable --aunque no imprescindible-- conseguir que los datos posean una serie de cualidades (Masters, 1993; Martín del Brío y Sanz, 1997; SPSS Inc., 1997; Sarle, 1998). Las variables deberían seguir una distribución normal o uniforme en tanto que el rango de posibles valores debería ser aproximadamente el mismo y acotado dentro del intervalo de trabajo de la función de activación empleada en las capas ocultas y de salida de la red neuronal.

Teniendo en cuenta lo comentado, las variables de entrada y salida suelen acotarse a valores comprendidos entre 0 y 1 ó entre -1 y 1. Si la variable es de naturaleza discreta, se utiliza la codificación dummy. Por ejemplo, la variable sexo podría codificarse como: 0 = hombre, 1 = mujer; estando representada por una única neurona. La variable nivel social podría codificarse como: 1 0 0 = bajo, 0 1 0 = medio, 0 0 1 = alto; estando representada por tres neuronas. Por su parte, si la variable es de naturaleza continua, ésta se representa mediante una sola neurona, como, por ejemplo, el CI de un sujeto.

3.2.- Creación de los conjuntos de aprendizaje, validación y test

En la metodología de las RNA, a fin de encontrar la red que tiene la mejor ejecución con casos nuevos --es decir, que sea capaz de generalizar--, la muestra de datos es a menudo subdividida en tres grupos (Bishop, 1995; Ripley, 1996): entrenamiento, validación y test.

Durante la etapa de aprendizaje de la red, los pesos son modificados de forma iterativa de acuerdo con los valores del grupo de entrenamiento, con el objeto de minimizar el error cometido entre la salida obtenida por la red y la salida deseada por el usuario. Sin embargo, como ya se ha comentado, cuando el número de parámetros o pesos es excesivo en relación al problema --fenómeno del sobreajuste--, el modelo se ajusta demasiado a las particularidades irrelevantes presentes en los patrones de entrenamiento en vez de ajustarse a la función subyacente que relaciona entradas y salidas, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos.

Para evitar el problema del sobreajuste, es aconsejable utilizar un segundo grupo de datos diferentes a los de entrenamiento, el grupo de validación, que permita controlar el proceso de aprendizaje. Durante el aprendizaje la red va modificando los pesos en función de los datos de entrenamiento y de forma alternada se va obteniendo el error

que comete la red ante los datos de validación. Este proceso se ve representado en la figura 3. Podemos observar cómo el error de entrenamiento y el error de validación van disminuyendo a medida que aumenta el número de iteraciones, hasta alcanzar un mínimo en la superficie del error, momento en el que podemos parar el aprendizaje de la red.

Con el grupo de validación podemos averiguar cuál es el número de pesos óptimo --y así evitar el problema del sobreajuste--, en función de la arquitectura que ha tenido la mejor ejecución con los datos de validación. Como se verá más adelante, mediante este grupo de validación también se puede determinar el valor de otros parámetros que intervienen en el aprendizaje de la red.

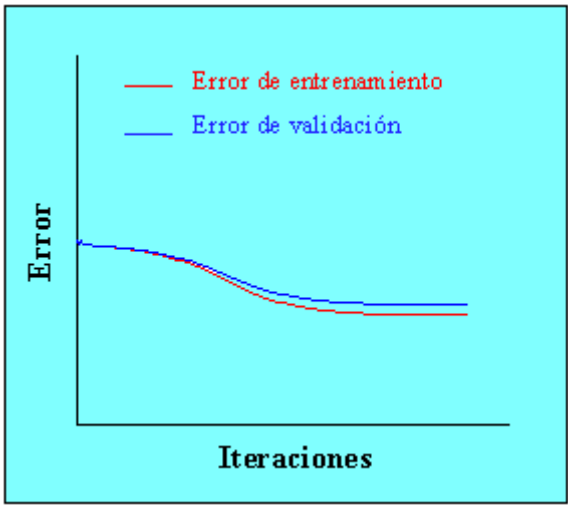


Figura 3: Evolución del error de entrenamiento y el error de validación.

Por último, si se desea medir de una forma completamente objetiva la eficacia final del sistema construido, no deberíamos basarnos en el error que se comete ante los datos de validación, ya que de alguna forma, estos datos han participado en el proceso de entrenamiento. Se debería contar con un tercer grupo de datos independientes, el grupo de test el cuál proporcionará una estimación insesgada del error de generalización.

3.3.- Entrenamiento de la red neuronal

Una vez visto el funcionamiento del algoritmo *backpropagation*, a continuación, se proporcionan una serie de consejos prácticos acerca de cuatro grupos de parámetros relacionados con el aprendizaje cuyo valor no se puede conocer *a priori* dado un

problema, sino que deben ser determinados mediante ensayo y error. La utilización de un grupo de validación ayudará a conocer el valor óptimo de cada uno de estos parámetros: valor de los pesos iniciales, arquitectura de la red, valor de la tasa de aprendizaje y del momento, y función de activación de las neuronas de la capa oculta y de salida. Así, la configuración de parámetros que obtenga el menor error ante los datos de validación, será la seleccionada para pasar a la fase de test.

3.3.1.- Elección de los pesos iniciales

Cuando una red neuronal es diseñada por primera vez, se deben asignar valores a los pesos a partir de los cuales comenzar la etapa de entrenamiento. Los pesos de umbral y de conexión se pueden inicializar de forma totalmente aleatoria, si bien es conveniente seguir algunas sencillas reglas que permitirán minimizar la duración del entrenamiento.

Es conveniente que la entrada neta a cada unidad sea cero, independientemente del valor que tomen los datos de entrada. En esta situación, el valor devuelto por la función de activación que se suele utilizar --la función sigmoidal--, es un valor intermedio, que proporciona el menor error si los valores a predecir se distribuyen simétricamente alrededor de este valor intermedio (como habitualmente sucede). Además, al evitar los valores de salida extremos se escapa de las zonas saturadas de la función sigmoidal en que la pendiente es prácticamente nula y, por tanto, el aprendizaje casi inexistente.

Para alcanzar este objetivo, la forma más sencilla y utilizada consiste en realizar una asignación de pesos pequeños generados de forma aleatoria, en un rango de valores entre -0.5 y 0.5 o algo similar (SPSS Inc., 1997).

3.3.2.- Arquitectura de la red

Respecto a la arquitectura de la red, se sabe que para la mayoría de problemas prácticos bastará con utilizar una sola capa oculta (Funahashi, 1989; Hornik, Stinchcombe y White, 1989).

El número de neuronas de la capa de entrada está determinado por el número de variables predictoras. Así, siguiendo los ejemplos de variables comentados en el apartado 3.1., la variable sexo estaría representada por una neurona que recibiría los

valores 0 ó 1. La variable estatus social estaría representada por tres neuronas que recibirían las codificaciones (1 0 0), (0 1 0) ó (0 0 1). Por último, la variable puntuación en CI estaría representada por una neurona que recibiría la puntuación previamente acotada, por ejemplo, a valores entre 0 y 1.

Por su parte, el número de neuronas de la capa de salida está determinado bajo el mismo esquema que en el caso anterior. Si estamos ante un problema de clasificación, cada neurona representará una categoría obteniendo un valor de activación máximo (por ejemplo, 1) la neurona que representa la categoría de pertenencia del patrón y un valor de activación mínimo (por ejemplo, 0) todas las demás neuronas de salida. Cuando intentamos discriminar entre dos categorías, bastará con utilizar una única neurona (por ejemplo, salida 1 para la categoría A, salida 0 para la categoría B). Si estamos ante un problema de estimación, tendremos una única neurona que dará como salida el valor de la variable a estimar.

Por último, el número de neuronas ocultas determina la capacidad de aprendizaje de la red neuronal. No existe una receta que indique el número óptimo de neuronas ocultas para un problema dado. Recordando el problema del sobreajuste, se debe usar el mínimo número de neuronas ocultas con las cuales la red rinda de forma adecuada (Masters, 1993; Smith, 1993; Rzempoluck, 1998). Esto se consigue evaluando el rendimiento de diferentes arquitecturas en función de los resultados obtenidos con el grupo de validación.

3.3.3.- Tasa de aprendizaje y factor momento

El valor de la tasa de aprendizaje (η) tiene un papel crucial en el proceso de entrenamiento de una red neuronal, ya que controla el tamaño del cambio de los pesos en cada iteración. Se deben evitar dos extremos: un ritmo de aprendizaje demasiado pequeño puede ocasionar una disminución importante en la velocidad de convergencia y la posibilidad de acabar atrapado en un mínimo local; en cambio, un ritmo de aprendizaje demasiado grande puede conducir a inestabilidades en la función de error, lo cual evitará que se produzca la convergencia debido a que se darán saltos en torno al mínimo sin alcanzarlo. Por tanto, se recomienda elegir un ritmo de aprendizaje lo más grande posible sin que provoque grandes oscilaciones. En general, el valor de la tasa de

aprendizaje suele estar comprendida entre 0.05 y 0.5, (Rumelhart, Hinton y Williams, 1986).

El factor momento (α) permite filtrar las oscilaciones en la superficie del error provocadas por la tasa de aprendizaje y acelera considerablemente la convergencia de los pesos, ya que si en el momento n el incremento de un peso era positivo y en n + 1 también, entonces el descenso por la superficie de error en n + 1 será mayor. Sin embargo, si en n el incremento era positivo y en n + 1 es negativo, el paso que se da en n + 1 es más pequeño, lo cual es adecuado, ya que eso significa que se ha pasado por un mínimo y los pasos deben ser menores para poder alcanzarlo. El factor momento suele tomar un valor próximo a 1 (por ejemplo, 0.9) (Rumelhart, Hinton y Williams, 1986).

3.3.4.- Función de activación de las neuronas ocultas y de salida

Hemos visto que para obtener el valor de salida de las neuronas de la capa oculta y de salida, se aplica una función, denominada función de activación, sobre la entrada neta de la neurona. El algoritmo *backpropagation* exige que la función de activación sea continua y, por tanto, derivable para poder obtener el error o valor delta de las neuronas ocultas y de salida. Se disponen de dos formas básicas que cumplen esta condición: la función lineal (o identidad) y la función sigmoidal (logística o tangente hiperbólica). En las figuras 4, 5 y 6 se presentan las expresiones matemáticas y la correspondiente representación gráfica de la función lineal, la sigmoidal logística (con límites entre 0 y 1) y la sigmoidal tangente hiperbólica (con límites entre -1 y 1):

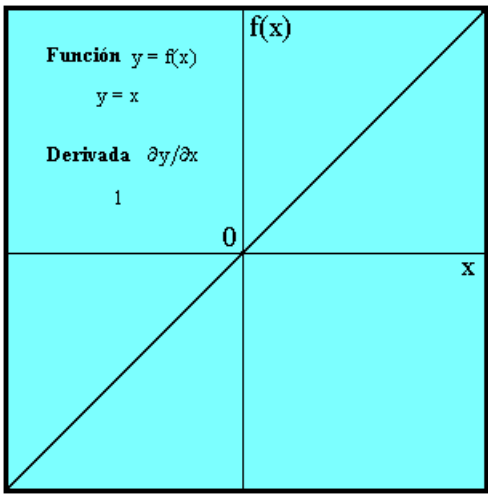


Figura 4: Función lineal.

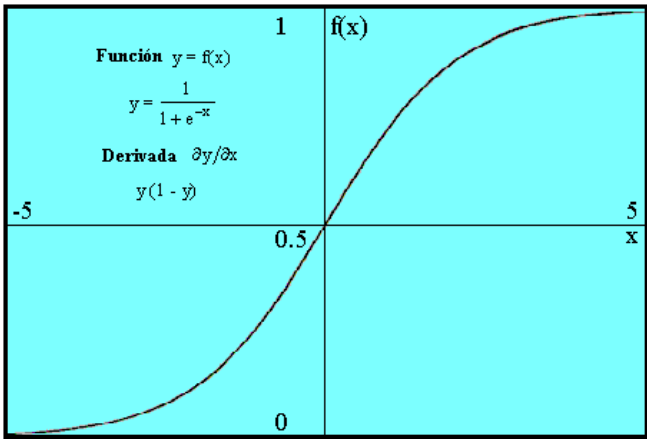


Figura 5: Función sigmoidal logística.

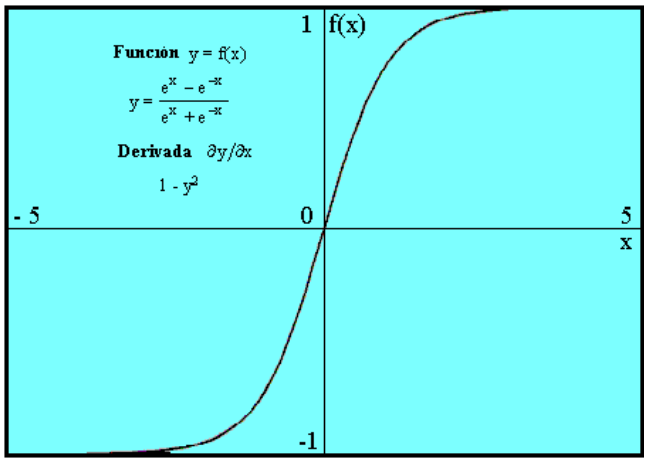


Figura 6: Función sigmoidal tangente hiperbólica.

Debemos tener en cuenta que para aprovechar la capacidad de las RNA de aprender relaciones complejas o no lineales entre variables, es absolutamente imprescindible la utilización de funciones no lineales al menos en las neuronas de la capa oculta (Rzempoluck, 1998). Las RNA que no utilizan funciones no lineales, se limitan a solucionar tareas de aprendizaje que implican únicamente funciones lineales o problemas de clasificación que son linealmente separables. Por tanto, en general se utilizará la función sigmoidal (logística o tangente hiperbólica) como función de activación en las neuronas de la capa oculta.

Por su parte, la elección de la función de activación en las neuronas de la capa de salida dependerá del tipo de tarea impuesto. En tareas de clasificación, las neuronas normalmente toman la función de activación sigmoidal. Así, cuando se presenta un patrón que pertenece a una categoría particular, los valores de salida tienden a dar como

valor 1 para la neurona de salida que representa la categoría de pertenencia del patrón, y 0 ó -1 para las otras neuronas de salida. En cambio, en tareas de predicción o aproximación de una función, generalmente las neuronas toman la función de activación lineal.

3.4.- Evaluación del rendimiento del modelo

Una vez seleccionado el modelo de red cuya configuración de parámetros ha obtenido la mejor ejecución ante el conjunto de validación, debemos evaluar la capacidad de generalización de la red de una forma completamente objetiva a partir de un tercer grupo de datos independiente, el conjunto de test.

Cuando la tarea de aprendizaje consiste en la estimación de una función, normalmente se utiliza la media cuadrática del error para evaluar la ejecución del modelo y viene dada por la siguiente expresión:

$$MC_{Error} = \frac{\sum_{p=1}^{P}\sum_{k=1}^{M}\left(d_{pk} - y_{pk}\right)^2}{P \cdot M}$$

Cuando se trata de un problema de clasificación de patrones es más cómodo basarnos en la frecuencia de clasificaciones correctas e incorrectas. A partir del valor de las frecuencias, podemos construir una tabla de confusión y calcular diferentes índices de asociación y acuerdo entre el criterio y la decisión tomada por la red neuronal. Por último, cuando estamos interesados en discriminar entre dos categorías, especialmente si utilizamos la red neuronal como instrumento diagnóstico (por ejemplo, salida = 0 -> sujeto sano; salida = 1 -> sujeto enfermo), es interesante hacer uso de los índices de sensibilidad, especificidad y eficacia, y del análisis de curvas ROC (*Receiver operating characteristic*) (Palmer, Montaño y Calafat, 2000).

3.5.- Interpretación de los pesos obtenidos

Una de las críticas más importantes que se han lanzado contra el uso de RNA trata sobre lo difícil que es comprender la naturaleza de las representaciones internas generadas por

la red para responder ante un problema determinado. A diferencia de los modelos estadísticos clásicos, no es tan evidente conocer en una red la importancia que tiene cada variable predictora sobre la salida del modelo. Sin embargo, esta percepción acerca de las RNA como una compleja "caja negra", no es del todo cierta. De hecho, han surgido diferentes intentos por interpretar los pesos de la red neuronal (Garson, 1991; Zurada, Malinowski y Cloete, 1994; Rambhia, Glenny y Hwang, 1999; Hunter, Kennedy, Henry et al., 2000), de los cuales el más ampliamente utilizado es el denominado análisis de sensibilidad.

El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida y_k debido al cambio que se produce en una entrada x_i. Así, cuanto mayor efecto se observe sobre la salida, mayor sensibilidad podemos deducir que presenta respecto a la entrada. Un método muy común para realizar este tipo de análisis consiste en fijar el valor de todas las variables de entrada a su valor medio e ir variando el valor de una de ellas a lo largo de todo su rango, registrando el valor de salida de la red. Este método se suele representar de forma gráfica utilizando una o dos variables de entrada sobre una de las salidas de la red. A modo de ejemplo, en la figura 7 se muestra la representación gráfica del análisis de sensibilidad a partir de dos variables de entrada sobre la salida de la red, habiendo fijado todas las demás variables de entrada a su valor medio.

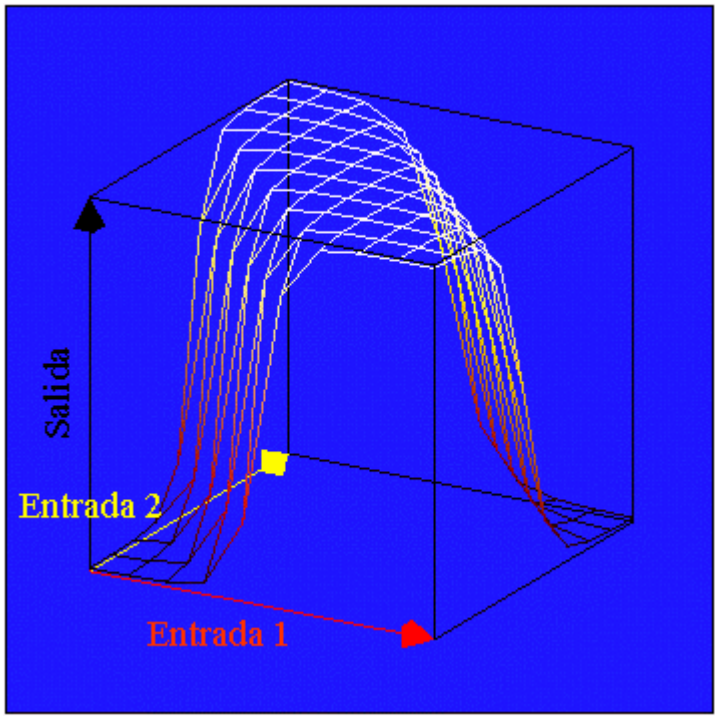


Figura 7: Representación gráfica del análisis de sensibilidad.

Este tipo de representación nos permite estudiar la forma que tiene la función que existe entre cada variable de entrada y cada variable de salida.

Otros autores han propuesto procedimientos numéricos para realizar el análisis de sensibilidad, como la obtención de la matriz jacobiana (Hwang, Choi, Oh et al., 1991; Fu y Chen, 1993; Bishop, 1995; Bahbah y Girgis, 1999). Veamos en qué consiste este procedimiento.

En la descripción del algoritmo *backpropagation*, hemos visto que la derivada parcial del error respecto a los pesos nos indica qué dirección tomar para modificar los pesos con el fin de reducir el error de forma iterativa. Mediante un procedimiento similar, los elementos que componen la matriz Jacobiana S proporcionan una medida de la sensibilidad de las salidas a cambios que se producen en cada una de las variables de entrada. En la matriz Jacobiana S –de orden KxI--, cada fila representa una salida de la red y cada columna representa una entrada de la red, de forma que el elemento S_{ki} de la matriz representa la sensibilidad de la salida k respecto a la entrada i. Cada uno de los elementos S_{ki} se obtiene calculando la derivada parcial de una salida y_k respecto a una entrada x_i, esto es:

$$\frac{\partial y_k}{\partial x_i}$$

Aplicando la regla de la cadena sobre esta expresión tenemos que:

$$S_{ki} = \frac{\partial y_k}{\partial x_i} = f'(net_k)\sum_{j=1}^{L} v_{kj} f'(net_j) w_{ji}$$

Así, cuanto mayor sea el valor de S_{ki}, más importante es x_i en relación a y_k. El signo de S_{ki} nos indicará si la relación entre ambas variables es directa o inversa.

4.- Recursos gratuitos en internet sobre el perceptron multicapa

En la Web podemos encontrar multitud de recursos relacionados con el campo de las RNA. A modo de ejemplos ilustrativos, describiremos el funcionamiento de dos applets

y un programa totalmente gratuitos que permiten simular el comportamiento de un perceptrón multicapa entrenado mediante el algoritmo *backpropagation*.

4.1.- Applets

Se han seleccionado dos ejemplos de applets, uno para demostrar el uso del perceptrón multicapa como clasificador de patrones, otro para demostrar el uso del perceptrón multicapa como estimador de funciones. El lector interesado en este tipo de recursos puede visitar la página del Instituto Nacional de Biociencia y Tecnología Humana (M.I.T.I.) de Japón, donde podrá encontrar un numeroso listado de applets demostrativos sobre RNA.

4.1.1.- Ejemplo de clasificación de patrones

Jason Tiscione ha desarrollado un applet denominado Reconocedor Óptico de Caracteres (OCHRE), el cual simula el comportamiento de un perceptrón multicapa aprendiendo a reconocer los dígitos del 0 al 9.

En la figura 8 se muestra la ventana del applet OCHRE:

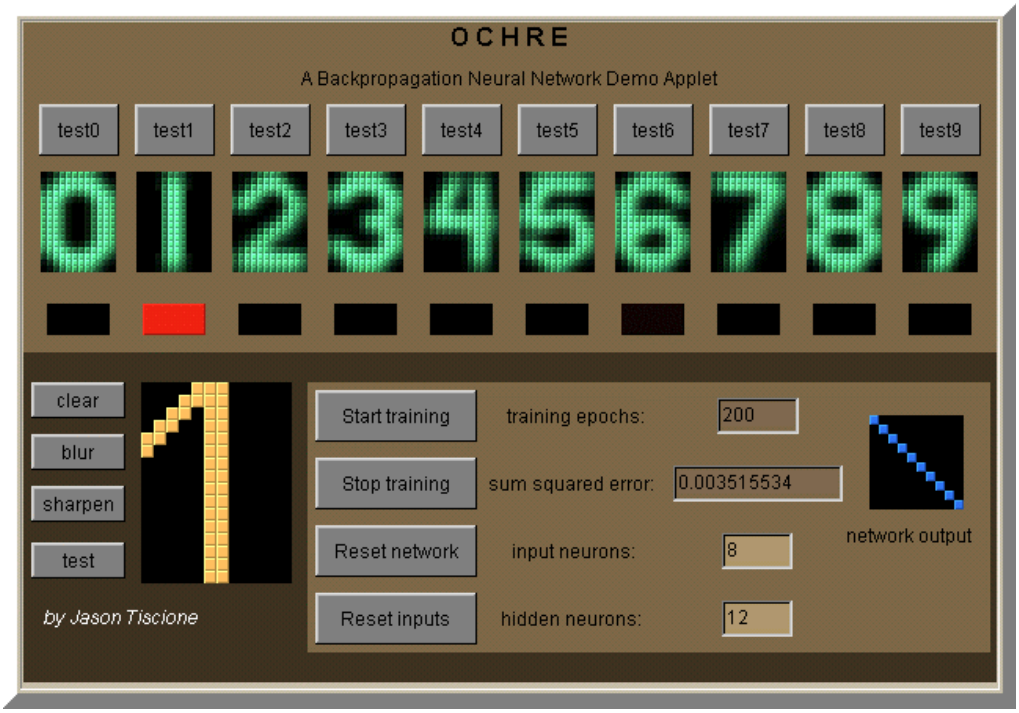


Figura 8: Ventana del applet OCHRE .

Los iconos que contienen los diez dígitos, situados a lo largo de la parte superior de la ventana del applet, son los patrones de entrada usados para entrenar y testar la red neuronal. La red está compuesta por ocho neuronas de entrada encargadas de recibir cada uno de los patrones, 12 neuronas ocultas (aunque este valor se puede manipular) y 10 neuronas de salida cada una de las cuales representa un dígito.

Para testar la ejecución de una red, pulse el botón "test" que aparece en la parte superior de cada uno de iconos. La fila situada debajo de los iconos indica el nivel de activación de las 10 neuronas de salida. Cuando la red está totalmente entrenada, la única neurona de salida activada será aquella que represente el icono que precisamente se esté probando. La línea situada debajo de tal icono aparecerá de color rojo señalando un nivel de activación máximo, mientras que las otras permanecerán en negro.

Para entrenar la red neuronal, pulse el botón "Start training" que aparece en la parte inferior de la ventana. Cuando el applet comienza el aprendizaje, se puede observar cómo se incrementa el número de iteraciones o épocas de entrenamiento, y cómo se reduce lentamente la suma cuadrática del error. También se puede observar cómo cambia de apariencia el icono de salida de la red (en la parte inferior derecha del applet). Este icono resume las respuestas dadas por la red ante los 10 patrones de entrenamiento, de forma que los cuadrados azules indican respuestas apropiadas y los cuadrados rojos indican respuestas inapropiadas.

El entrenamiento tarda pocos minutos y se completa después de 150-250 iteraciones, cuando la suma cuadrática del error alcanza un valor bajo (aproximadamente 0.01). En este punto se puede pulsar el botón "Stop training" para parar el entrenamiento de la red. Si, a continuación, se pulsa el botón "test" situado encima de cada dígito, la línea situada debajo del icono apropiado se activará.

El usuario también puede testar la red neuronal dibujando un dígito con el ratón en el icono situado en la parte inferior izquierda del applet. Con el botón izquierdo del ratón se dibuja un pixel, mientras que con el botón derecho se puede borrar un pixel. Los botones "blur" y "sharpen" situados a la izquierda del icono de dibujo, permiten hacer los trazos más o menos definidos. Para averiguar qué "piensa" la red neuronal acerca del dibujo realizado, pulse el botón "test" situado en la parte inferior izquierda.

La red puede ser entrenada a partir de patrones de entrada creados por el propio usuario. Se puede alterar cualquier dígito situado en la parte superior, dibujando directamente sobre él mediante los botones del ratón. Para definir, añadir ruido o limpiar el dibujo, use las teclas "B", "S" y "C", respectivamente. Para restaurar los dígitos originales, pulse el botón "Reset inputs". Para restaurar la red entera, pulse el botón "Reset network".

Se puede cambiar el número de neuronas en la capa de entrada y oculta modificando el número en el campo correspondiente antes de apretar el botón "Reset network". Si determinamos pocas neuronas ocultas, la red será incapaz de aprender la tarea. Con demasiadas neuronas ocultas, es probable que se produzca un sobreajuste, provocando una reducción en la capacidad de generalización del modelo.

4.1.2.- Ejemplo de aproximación de funciones

Paul Watta, Mohamad Hassoun y Norman Dannug han desarrollado una Herramienta de Aprendizaje Backpropagation para la Aproximación de Funciones.

En la figura 9 se muestra la ventana del applet:

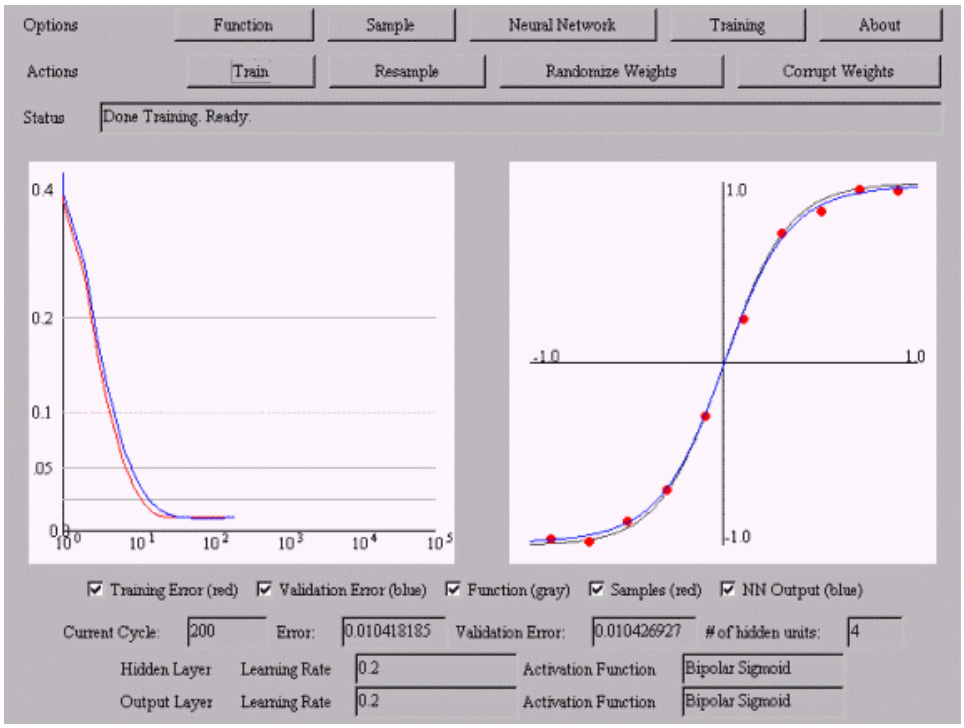


Figura 9: Ventana del applet para la aproximación de funciones.

Esta herramienta tiene una serie de opciones interesantes situadas en la parte superior. En este sentido, pulsando el botón "Function", podemos determinar el tipo de función que queremos que aprenda la red neuronal (por ejemplo, cuadrática, sigmoidal, exponencial, etc.). El botón "Sample" nos permite fijar el número de patrones de entrenamiento y validación. Con el botón "Neural Network" podemos determinar diferentes parámetros relacionados con la red neuronal: número de neuronas ocultas, tasa de aprendizaje para las neuronas de la capa oculta y de salida, y función de activación para las neuronas de la capa oculta y de salida. Por último, el botón "Training" nos permite fijar el número total de iteraciones o ciclos de aprendizaje y el valor del error objetivo (*target*).

Por otra parte, podemos realizar una serie de acciones situadas también en la parte superior. En este sentido, pulsando el botón "Training", comenzamos el entrenamiento de la red neuronal. Con el botón "Resample" se crea aleatoriamente un nuevo juego de patrones de entrenamiento y validación. El botón "Randomize Weights" nos permite crear una nueva configuración inicial de pesos. Por último, con el botón "Corrupt Weights" podemos introducir ruido a los pesos de la red.

Esta herramienta cuenta con dos tipos de representación gráfica. Por una parte, se muestra la evolución del error de entrenamiento y el error de validación a lo largo del aprendizaje de la red. Por otra parte, se muestra en un plano bidimensional la forma de la función que se pretende aprender, la situación de los patrones de entrenamiento y la salida de la red en el plano.

Por último, en la parte inferior se muestra una serie de informaciones acerca de la red neuronal a lo largo del proceso de aprendizaje. Así, podemos ver el número de iteraciones o ciclos de entrenamiento realizados hasta el momento, el error de entrenamiento y el error de validación, el número de neuronas ocultas, el valor de la tasa de aprendizaje utilizada en la capa oculta y de salida, y la función de activación de las neuronas ocultas y de salida.

4.2.- Software

En la Web podemos encontrar multitud de programas simuladores de redes neuronales de libre distribución (gratuitos o *sharewares*). El lector interesado puede visitar dos

listados completos sobre este tipo de programas: Por un lado, tenemos el listado ofrecido por el grupo de noticias sobre redes neuronales comp.ai.neural-nets, por otro lado, tenemos el listado ofrecido por el Pacific Northwest National Laboratory.

4.2.1.- Programa QwikNet

De entre los simuladores de libre distribución que podemos encontrar en internet, cabe destacar un programa *shareware* (para Windows 95/98/NT 4.0), el QwikNet 2.23 desarrolllado por Craig Jensen. Se trata de un simulador del perceptrón multicapa sencillo de manejar, que ayudará al lector a comprender mejor todos los conceptos expuestos a lo largo del documento.

Con la versión *shareware* se puede utilizar un máximo de 10 neuronas en una capa oculta y 500 patrones de entrenamiento. La versión registrada no tiene ningún tipo de limitación (se pueden utilizar hasta cinco capas ocultas).

Para bajarse la versión *shareware* del QwikNet pulse el siguiente botón:

El archivo bajado es un archivo comprimido (.ZIP). Para instalar el programa, descomprima el contenido del archivo .ZIP en una carpeta (mediante el programa WinZip) y ejecute el archivo setup.exe, el cual le guiará de forma sencilla a lo largo del proceso de instalación de QwikNet.

El programa QwikNet cuenta con un archivo de ayuda muy completo, así que nos limitaremos a describir brevemente las opciones más sobresalientes con las que cuenta el simulador.

Como se puede observar en la figura 10, la ventana del programa se divide en un conjunto de opciones agrupadas en secciones:

La sección *Training Properties* permite fijar los valores de la tasa de aprendizaje (eta) y el factor momento (alpha).

En la sección *Stopping Criteria* podemos determinar diferentes criterios de parada del entrenamiento de la red (número de iteraciones o épocas, valores de error y porcentaje de clasificaciones correctas para los patrones de entrenamiento y test).

La sección *Training Algorithm* permite seleccionar el algoritmo de aprendizaje. Así, tenemos el *On line backpropagation*, el *On line backpropagation* con el orden de los patrones aleatorizado, el *batch backpropagation*, el *delta-bar-delta*, el RPROP y el *QuickProp*.

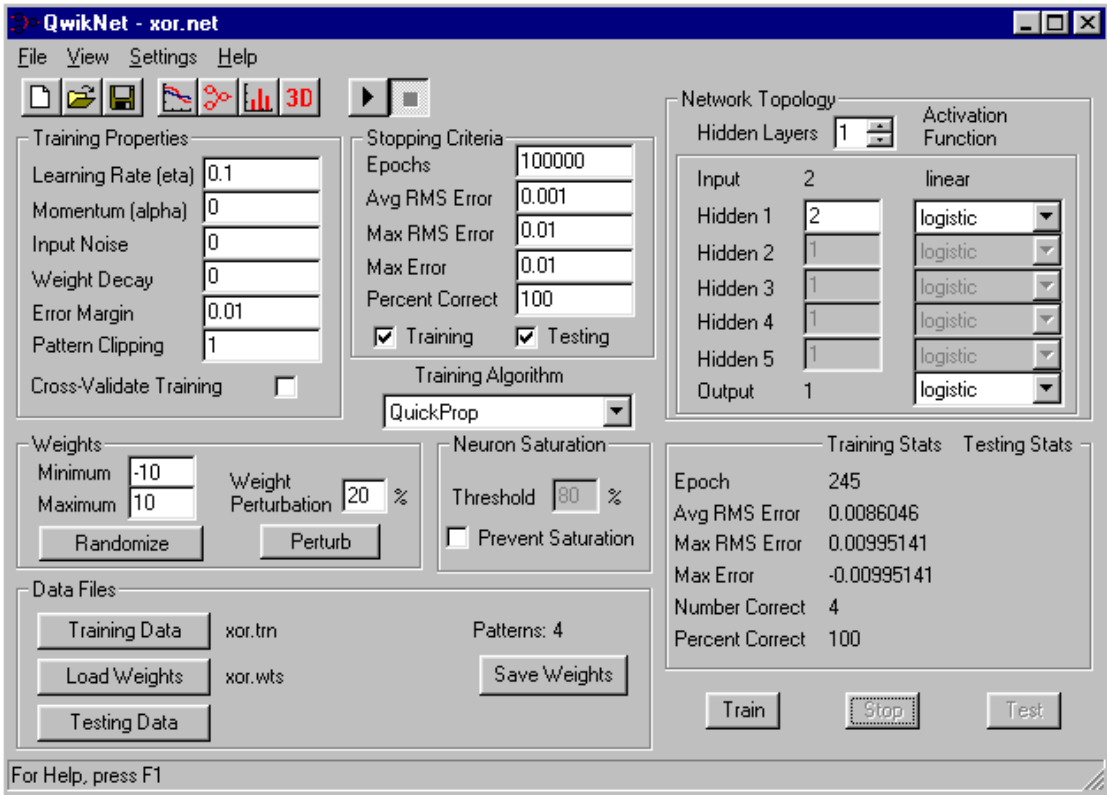


Figura 10: Ventana del programa QwikNet.

En la sección *Network Topology* podemos determinar el número de capas ocultas (hasta cinco capas), el número de neuronas por capa oculta y la función de activación de las neuronas ocultas y de salida (sigmoidal logística, sigmoidal tangente hiperbólica, lineal y gaussiana).

La sección *Weights* permite fijar el valor máximo y mínimo que puede adoptar un peso, e inicializar los pesos de la red de forma aleatoria en cualquier momento durante el proceso de aprendizaje.

En la sección *Data Files* podemos determinar los archivos que contienen los patrones de entrenamiento y test.

Durante el proceso de entrenamiento, las secciones *Training Stats* y *Testing Stats* muestran información acerca del número de ciclos de aprendizaje completados hasta el momento y el rendimiento del modelo ante los patrones de entrenamiento y test.

Los botones "Train", "Stop" y "Test" permiten comenzar el entrenamiento, pararlo y proporcionar la salida de la red ante los patrones de test, respectivamente.

Por último, el menú *View* situado en la parte superior de la ventana permite visualizar diferentes representaciones gráficas. Así, la opción *Network* muestra la arquitectura de la red, donde el color de las conexiones entre neuronas indica la magnitud de los pesos. La opción *Training Error Plot* permite ver la evolución del error de entrenamiento y test a medida que avanza el entrenamiento. La opción *Contour Plot* representa de forma gráfica el análisis de sensibilidad de una o dos variables de entrada sobre una variable de salida. La opción *Network Analysis Plot* representa mediante diagramas de barras el error cometido y las salidas proporcionadas por la red neuronal ante los patrones de entrenamiento y test.

Referencias bibliográficas

Arana, E., Delicado, P., Martí-Bonmatí, L. (1999). Validation procedures in radiologic diagnostic models. Neural network and logistic regression. *Investigative Radiology, 34*(10), 636-642.

Arbib, M.A. (Ed.) (1995). *The handbook of brain theory and neural networks*. Cambridge, Mass.: MIT Press.

Arbib, M.A., Erdi, P. y Szentagothai, J. (1997). *Neural organization: structure, function and dynamics*. Cambridge, Mass.: MIT Press.

Bahbah, A.G. y Girgis, A.A. (1999). Input feature selection for real-time transient stability assessment for artificial neural network (ANN) using ANN sensitivity analysis. *Proceedings of the 21 st International Conference on Power Industry Computer Applications*, 295-300.

Battiti, R. (1992). First and second order methods for learning: between steepest descent and Newton's method. *Neural Computation, 4*(2), 141-166.

Bishop, C.M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

De Lillo, A. y Meraviglia, C. (1998). The role of social determinants on men´s and women´s mobility in Italy. A comparison of discriminant analysis and artificial neural networks. *Substance Use and Misuse, 33*(3), 751-764.

Fahlman, S.E. (1988). Faster-learning variations on back-propagation: an empirical study. *Proceedings of the 1988 Connectionist Models Summer School*, 38-51.

Fu, L. y Chen, T. (1993). Sensitivity analysis for input vector in multilayer feedforward neural networks. *Proceedings of IEEE International conference on Neural Networks*, 215-218.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural networks. *Neural Networks, 2*, 183-192.

Garson, G.D. (1991). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.

Hornik, K., Stinchcombe, M. y White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

Hunter, A., Kennedy, L., Henry, J. y Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine*, *62*, 11-19.

Hwang, J.N., Choi, J.J., Oh, S. y Marks, R.J. (1991). Query based learning applied to partially trained multilayer perceptron. *IEEE T-NN*, *2*(1), 131-136.

Jacobs, R.A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, *1*(4), 295-308.

Jang, J. (1998). Comparative analysis of statistical methods and neural networks for predicting life insurers´ insolvency (bankruptcy) (The University of Texas at Austin, 1997). *Dissertation Abstracts International, DAI-A, 59/01*, 228.

Le Cun, Y. (1985). *Modèles connexionnistes de l'apprentissage*. Tesis doctoral. Université Pierre et Marie Curie, París VI.

Martín del Brío, B. y Sanz, A. (1997). *Redes neuronales y sistemas borrosos*. Madrid: Ra-Ma.

Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press.

Palmer, A. y Montaño, J.J. (1999). ¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones. *Adicciones, 11*(3), 243-255.

Palmer, A., Montaño, J.J. y Calafat, A. (2000). Predicción del consumo de éxtasis a partir de redes neuronales artificiales. *Adicciones, 12*(1), 29-41.

Parker, D. (1982). *Learning logic*. Invention Report, S81-64, File 1. Office of Technology Lisensing, Stanford University.

Rambhia, A.H., Glenny, R. y Hwang, J. (1999). Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1097-1100.

Riedmiller, M. y Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *IEEE International Conference on Neural Networks*, 586-591.

Ripley, B.D. (1996). *Pattern recognition and neural networks*. Cambridge: Cambridge University Press.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Rzempoluck, E.J. (1998). *Neural network data analysis using Simulnet*. New York: Springer-Verlag.

Sarle, W.S. (Ed.) (1998). *Neural network FAQ*. Periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: ftp://ftp.sas.com/pub/neural/FAQ.html.

Simpson, P.K. (Ed.) (1995). *Neural networks technology and applications: theory, technology and implementations*. New York: IEEE.

Smith, M. (1993). *Neural networks for statistical modeling*. New York: Van Nostrand Reinhold.

SPSS Inc. (1997). *Neural Connection 2.0: User´s Guide* [Manual de programa para ordenadores]. Chicago: SPSS Inc.

Takahashi, K., Hayasawa, H. y Tomita, M. (1999). A predictive model for affect of atopic dermatitis in infancy by neural network and multiple logistic regression. *Japanese Journal of Allergology, 48*(11), 1222-1229.

Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psycometrika, 63*(1), 5-22.

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Tesis doctoral. Harvard University.

Zurada, J.M., Malinowski, A. y Cloete, I. (1994). Sensitivity analysis for minimization of input data dimension for feedforward neural network. *Proceedings of IEEE International Symposium on Circuits and Systems*, 447-450.

Accesible en Internet desde el 9/3/2001
http://www.psiquiatria.com/psicologia/revista/61/2833

Tutorial sobre redes neuronales artificiales: los mapas autoorganizados de Kohonen.

REVISTA ELECTRÓNICA DE PSICOLOGÍA
Vol. 6, No. 1, Enero 2002
ISSN 1137-8492

Tutorial sobre Redes Neuronales Artificiales: Los Mapas Autoorganizados de Kohonen

Palmer, A., Montaño, J.J. y Jiménez, R.
Área de Metodología de las Ciencias del Comportamiento.
Facultad de Psicología.
Universitat de les Illes Balears.
e-Mail: alfonso.palmer@uib.es

1.- Introducción

En los últimos quince años, las redes neuronales artificiales (RNA) han emergido como una potente herramienta para el modelado estadístico orientada principalmente al reconocimiento de patrones –tanto en la vertiente de clasificación como de predicción. Las RNA poseen una serie de características admirables, tales como la habilidad para procesar datos con ruido o incompletos, la alta tolerancia a fallos que permite a la red operar satisfactoriamente con neuronas o conexiones dañadas y la capacidad de responder en tiempo real debido a su paralelismo inherente.

Actualmente, existen unos 40 paradigmas de RNA que son usados en diversos campos de aplicación (Taylor, 1996; Arbib, Erdi y Szentagothai, 1997; Sarle, 1998). Entre estos paradigmas, podemos destacar la red *backpropagation* (Rumelhart, Hinton y Williams, 1986) y los mapas autoorganizados de Kohonen (Kohonen, 1982a, 1982b).

La red *backpropagation*, mediante un esquema de aprendizaje supervisado, ha sido utilizada satisfactoriamente en la clasificación de patrones y la estimación de funciones. La descripción de este tipo de red se puede encontrar en un documento anterior (Palmer, Montaño y Jiménez, en prensa).

En el presente documento, nos proponemos describir otro de los sistemas neuronales más conocidos y empleados, los mapas autoorganizados de Kohonen. Este tipo de red neuronal, mediante un aprendizaje no supervisado, puede ser de gran utilidad en el campo del análisis exploratorio de datos, debido a que son sistemas capaces de realizar análisis de clusters, representar densidades de probabilidad y proyectar un espacio de alta dimensión sobre otro de dimensión mucho menor. A fin de asegurar la comprensión de los conceptos expuestos por parte del lector y, al mismo tiempo, explotar al máximo los recursos que nos ofrece la Web, tal descripción irá acompañada de applets y software ilustrativos, los cuales estarán a disposición del lector via internet.

2.- Los mapas autoorganizados de Kohonen

En 1982 Teuvo Kohonen presentó un modelo de red denominado mapas autoorganizados o SOM (*Self-Organizing Maps*), basado en ciertas evidencias descubiertas a nivel cerebral y con un gran potencial de aplicabilidad práctica. Este tipo

de red se caracteriza por poseer un aprendizaje no supervisado competitivo. Vamos a ver en qué consiste este tipo de aprendizaje.

A diferencia de lo que sucede en el aprendizaje supervisado, en el no supervisado (o autoorganizado) no existe ningún maestro externo que indique si la red neuronal está operando correcta o incorrectamente, pues no se dispone de ninguna salida objetivo hacia la cual la red neuronal deba tender. Así, durante el proceso de aprendizaje la red autoorganizada debe descubrir por sí misma rasgos comunes, regularidades, correlaciones o categorías en los datos de entrada, e incorporarlos a su estructura interna de conexiones. Se dice, por tanto, que las neuronas deben autoorganizarse en función de los estímulos (datos) procedentes del exterior.

Dentro del aprendizaje no supervisado existe un grupo de modelos de red caracterizados por poseer un aprendizaje competitivo. En el aprendizaje competitivo las neuronas compiten unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje, se pretende que cuando se presente a la red un patrón de entrada, sólo una de las neuronas de salida (o un grupo de vecinas) se active. Por tanto, las neuronas compiten por activarse, quedando finalmente una como neurona vencedora y anuladas el resto, que son forzadas a sus valores de respuesta mínimos.

El objetivo de este aprendizaje es categorizar (clusterizar) los datos que se introducen en la red. De esta forma, las informaciones similares son clasificadas formando parte de la misma categoría y, por tanto, deben activar la misma neurona de salida. Las clases o categorías deben ser creadas por la propia red, puesto que se trata de un aprendizaje no supervisado, a través de las correlaciones entre los datos de entrada.

2.1.- Fundamentos biológicos

Se ha observado que en el córtex de los animales superiores aparecen zonas donde las neuronas detectoras de rasgos se encuentran topológicamente ordenadas (Kohonen, 1989, 1990); de forma que las informaciones captadas del entorno a través de los órganos sensoriales, se representan internamente en forma de mapas bidimensionales. Por ejemplo, en el área somatosensorial, las neuronas que reciben señales de sensores que se encuentran próximos en la piel se sitúan también próximas en el córtex, de manera que reproducen --de forma aproximada--, el mapa de la superficie de la piel en

una zona de la corteza cerebral. En el sistema visual se han detectado mapas del espacio visual en zonas del cerebro. Por lo que respecta al sentido del oído, existen en el cerebro áreas que representan mapas tonotópicos, donde los detectores de determinados rasgos relacionados con el tono de un sonido se encuentran ordenados en dos dimensiones (Martín del Brío y Sanz, 1997).

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje. Esto sugiere, por tanto, que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior (Kohonen, 1982a).

Por otra parte, también se ha observado que la influencia que una neurona ejerce sobre las demás es función de la distancia entre ellas, siendo muy pequeña cuando están muy alejadas. Así, se ha comprobado que en determinados primates se producen interacciones laterales de tipo excitatorio entre neuronas próximas en un radio de 50 a 100 micras, de tipo inhibitorio en una corona circular de 150 a 400 micras de anchura alrededor del círculo anterior, y de tipo excitatorio muy débil, prácticamente nulo, desde ese punto hasta una distancia de varios centímetros. Este tipo de interacción tiene la forma típica de un sombrero mejicano como veremos más adelante.

En base a este conjunto de evidencias, el modelo de red autoorganizado presentado por Kohonen pretende mimetizar de forma simplificada la capacidad del cerebro de formar mapas topológicos a partir de las señales recibidas del exterior.

2.2.- Arquitectura

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada (formada por N neuronas, una por cada variable de entrada) se encarga de recibir y transmitir a la capa de salida la información procedente del exterior. La capa de salida (formada por M neuronas) es la encargada de procesar la información y formar el mapa de rasgos. Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional como se muestra en la figura 1, aunque a veces también se utilizan capas de una sola dimensión (cadena lineal de neuronas) o de tres dimensiones (paralelepípedo).

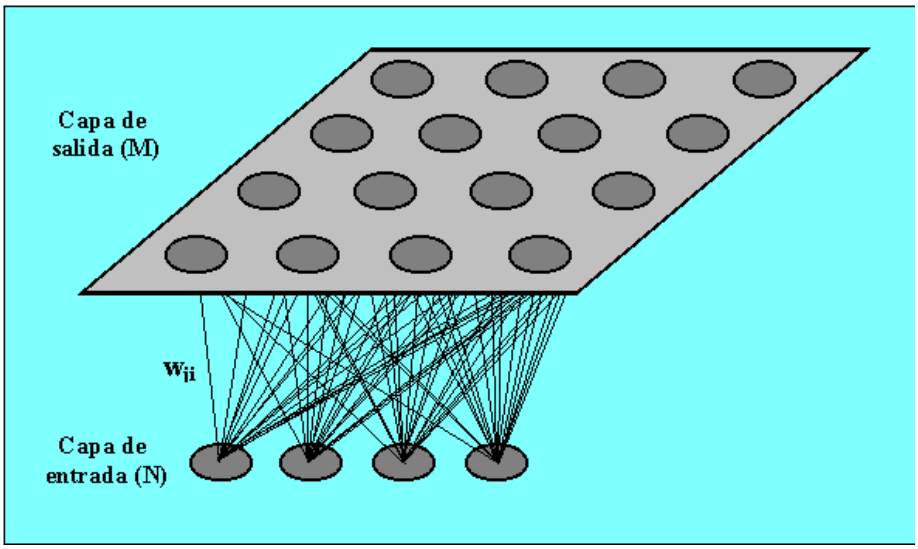


Figura 1: Arquitectura del SOM.

Las conexiones entre las dos capas que forman la red son siempre hacia delante, es decir, la información se propaga desde la capa de entrada hacia la capa de salida. Cada neurona de entrada i está conectada con cada una de las neuronas de salida j mediante un peso w_{ji}. De esta forma, las neuronas de salida tienen asociado un vector de pesos W_j llamado vector de referencia (o *codebook*), debido a que constituye el vector prototipo (o promedio) de la categoría representada por la neurona de salida j.

Entre las neuronas de la capa de salida, puede decirse que existen conexiones laterales de excitación e inhibición implícitas, pues aunque no estén conectadas, cada una de estas neuronas va a tener cierta influencia sobre sus vecinas. Esto se consigue a través de un proceso de competición entre las neuronas y de la aplicación de una función denominada de vecindad como veremos más adelante.

2.3.- Algoritmo

En el algoritmo asociado al modelo SOM podemos considerar, por un lado, una etapa de funcionamiento donde se presenta, ante la red entrenada, un patrón de entrada y éste se asocia a la neurona o categoría cuyo vector de referencia es el más parecido y, por otro lado, una etapa de entrenamiento o aprendizaje donde se organizan las categorías que forman el mapa mediante un proceso no supervisado a partir de las relaciones descubiertas en el conjunto de los datos de entrenamiento.

2.3.1.- Etapa de funcionamiento

Cuando se presenta un patrón p de entrada X_p: $x_{p1},..., x_{pi},...,x_{pN}$, éste se transmite directamente desde la capa de entrada hacia la capa de salida. En esta capa, cada neurona calcula la similitud entre el vector de entrada X_p y su propio vector de pesos W_j o vector de referencia según una cierta medida de distancia o criterio de similitud establecido. A continuación, simulando un proceso competitivo, se declara vencedora la neurona cuyo vector de pesos es el más similar al de entrada.

La siguiente expresión matemática representa cuál de las M neuronas se activará al presentar el patrón de entrada X_p:

$$y_{pj} = \begin{cases} 1 & \min \left\| X_p - W_j \right\| \\ 0 & \text{resto} \end{cases}$$

donde y_{pj} representa la salida o el grado de activación de las neuronas de salida en función del resultado de la competición (1 = neurona vencedora, 0 = neurona no vencedora), $||X_p-W_j||$ representa una medida de similitud entre el vector o patrón de entrada X_p: $x_{p1},..., x_{pi},...,x_{pN}$ y el vector de pesos W_j: $w_{j1},..., w_{ji},...w_{jN}$, de las conexiones entre cada una de las neuronas de entrada y la neurona de salida j. En el siguiente apartado veremos las medidas de similitud más comúnmente utilizadas. En cualquier caso, la neurona vencedora es la que presenta la diferencia mínima.

En esta etapa de funcionamiento, lo que se pretende es encontrar el vector de referencia más parecido al vector de entrada para averiguar qué neurona es la vencedora y, sobre todo, en virtud de las interacciones excitatorias e inhibitorias que existen entre las neuronas, para averiguar en qué zona del espacio bidimensional de salida se encuentra tal neurona. Por tanto, lo que hace la red SOM es realizar una tarea de clasificación, ya que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información de entrada. Además, como ante otra entrada parecida se activa la misma neurona de salida, u otra cercana a la anterior, debido a la semejanza entre las clases, se garantiza que las neuronas topológicamente próximas sean sensibles a entradas físicamente similares. Por este motivo, la red es especialmente útil para

establecer relaciones, desconocidas previamente, entre conjuntos de datos (Hilera y Martínez, 1995).

2.3.2.- Etapa de aprendizaje

Se debe advertir, en primer lugar, que no existe un algoritmo de aprendizaje totalmente estándar para la red SOM. Sin embargo, se trata de un procedimiento bastante robusto ya que el resultado final es en gran medida independiente de los detalles de su realización concreta. En consecuencia, trataremos de exponer el algoritmo más habitual asociado a este modelo (Kohonen, 1982a, 1982b, 1989, 1995).

El algoritmo de aprendizaje trata de establecer, mediante la presentación de un conjunto de patrones de entrenamiento, las diferentes categorías (una por neurona de salida) que servirán durante la etapa de funcionamiento para realizar clasificaciones de nuevos patrones de entrada.

De forma simplificada, el proceso de aprendizaje se desarrolla de la siguiente manera. Una vez presentado y procesado un vector de entrada, se establece a partir de una medida de similitud, la neurona vencedora, esto es, la neurona de salida cuyo vector de pesos es el más parecido respecto al vector de entrada. A continuación, el vector de pesos asociado a la neurona vencedora se modifica de manera que se parezca un poco más al vector de entrada. De este modo, ante el mismo patrón de entrada, dicha neurona responderá en el futuro todavía con más intensidad. El proceso se repite para un conjunto de patrones de entrada los cuales son presentados repetidamente a la red, de forma que al final los diferentes vectores de pesos sintonizan con uno o varios patrones de entrada y, por tanto, con dominios específicos del espacio de entrada. Si dicho espacio está dividido en grupos, cada neurona se especializará en uno de ellos, y la operación esencial de la red se podrá interpretar como un análisis de clusters.

La siguiente interpretación geométrica (Masters, 1993) del proceso de aprendizaje puede resultar interesante para comprender la operación de la red SOM. El efecto de la regla de aprendizaje no es otro que acercar de forma iterativa el vector de pesos de la neurona de mayor actividad (ganadora) al vector de entrada. Así, en cada iteración el vector de pesos de la neurona vencedora rota hacia el de entrada, y se aproxima a él en una cantidad que depende del tamaño de una tasa de aprendizaje.

En la figura 2 se muestra cómo opera la regla de aprendizaje para el caso de varios patrones pertenecientes a un espacio de entrada de dos dimensiones, representados en la figura por los vectores de color negro. Supongamos que los vectores del espacio de entrada se agrupan en tres clusters, y supongamos que el número de neuronas de la red es también tres. Al principio del entrenamiento los vectores de pesos de las tres neuronas (representados por vectores de color rojo) son aleatorios y se distribuyen por la circunferencia. Conforme avanza el aprendizaje, éstos se van acercando progresivamente a las muestras procedentes del espacio de entrada, para quedar finalmente estabilizados como centroides de los tres clusters.

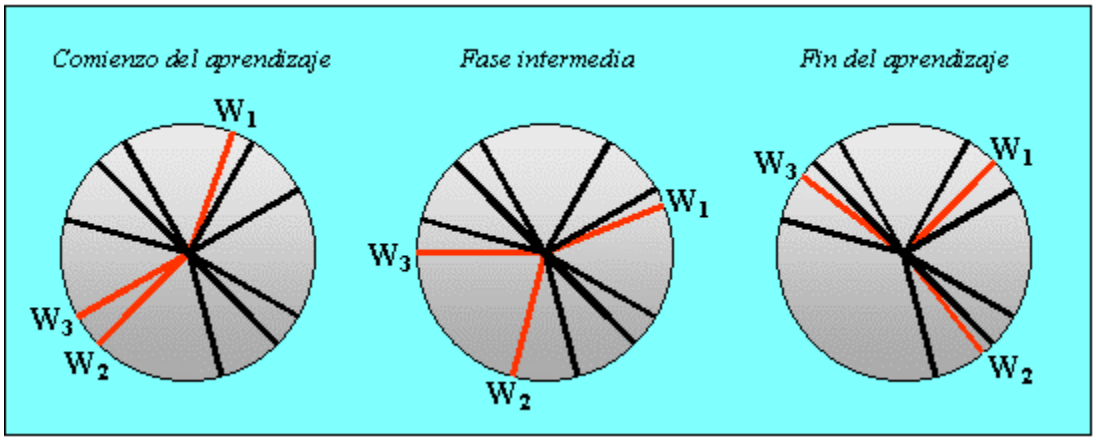


Figura 2: Proceso de aprendizaje en dos dimensiones.

Al finalizar el aprendizaje, el vector de referencia de cada neurona de salida se corresponderá con el vector de entrada que consigue activar la neurona correspondiente. En el caso de existir más patrones de entrenamiento que neuronas de salida, como en el ejemplo expuesto, más de un patrón deberá asociarse con la misma neurona, es decir, pertenecerán a la misma clase. En tal caso, los pesos que componen el vector de referencia se obtienen como un promedio (centroide) de dichos patrones.

Además de este esquema de aprendizaje competitivo, el modelo SOM aporta una importante novedad, pues incorpora relaciones entre las neuronas próximas en el mapa. Para ello, introduce una función denominada zona de vecindad que define un entorno alrededor de la neurona ganadora actual (vecindad); su efecto es que durante el aprendizaje se actualizan tanto los pesos de la vencedora como los de las neuronas pertenecientes a su vecindad. De esta manera, en el modelo SOM se logra que neuronas

próximas sintonicen con patrones similares, quedando de esta manera reflejada sobre el mapa una cierta imagen del orden topológico presente en el espacio de entrada.

Una vez entendida la forma general de aprendizaje del modelo SOM, vamos a expresar este proceso de forma matemática. Recordemos que cuando se presenta un patrón de entrenamiento, se debe identificar la neurona de salida vencedora, esto es, la neurona cuyo vector de pesos sea el más parecido al patrón presentado. Un criterio de similitud muy utilizado es la distancia euclídea que viene dado por la siguiente expresión:

$$\min \left\| X_p - W_j \right\| = \min \sum_{i=1}^{N} (x_{pi} - w_{ji})^2$$

De acuerdo con este criterio, dos vectores serán más similares cuanto menor sea su distancia.

Una medida de similitud alternativa más simple que la euclídea, es la correlación o producto escalar:

$$\min \left\| X_p - W_j \right\| = \max \sum_{i=1}^{N} x_{pi} \cdot w_{ji}$$

según la cual, dos vectores serán más similares cuanto mayor sea su correlación.

Identificada la neurona vencedora mediante el criterio de similitud, podemos pasar a modificar su vector de pesos asociado y el de sus neuronas vecinas, según la regla de aprendizaje:

$$\Delta w_{ji}(n+1) = \alpha(n)\left(x_{pi} - w_{j*i}(n)\right) \quad \text{para } j \in \text{Zona}_{j*}(n)$$

donde n hace referencia al número de ciclos o iteraciones, esto es, el número de veces que ha sido presentado y procesado todo el juego de patrones de entrenamiento. $\alpha(n)$ es la tasa de aprendizaje que, con un valor inicial entre 0 y 1, decrece con el número de iteraciones (n) del proceso de aprendizaje. $\text{Zona}_{j*}(n)$ es la zona de vecindad alrededor de la neurona vencedora j* en la que se encuentran las neuronas cuyos pesos son

actualizados. Al igual que la tasa de aprendizaje, el tamaño de esta zona normalmente se va reduciendo paulatinamente en cada iteración, con lo que el conjunto de neuronas que pueden considerarse vecinas cada vez es menor.

Tradicionalmente el ajuste de los pesos se realiza después de presentar cada vez un patrón de entrenamiento, como se muestra en la regla de aprendizaje expuesta. Sin embargo, hay autores (Masters, 1993) que recomiendan acumular los incrementos calculados para cada patrón de entrenamiento y, una vez presentados todos los patrones, actualizar los pesos a partir del promedio de incrementos acumulados. Mediante este procedimiento se evita que la dirección del vector de pesos vaya oscilando de un patrón a otro y acelera la convergencia de los pesos de la red.

En el proceso general de aprendizaje suelen considerarse dos fases. En la primera fase, se pretende organizar los vectores de pesos en el mapa. Para ello, se comienza con una tasa de aprendizaje y un tamaño de vecindad grandes, para luego ir reduciendo su valor a medida que avanza el aprendizaje. En la segunda fase, se persigue el ajuste fino del mapa, de modo que los vectores de pesos se ajusten más a los vectores de entrenamiento. El proceso es similar al anterior aunque suele ser más largo, tomando la tasa de aprendizaje constante e igual a un pequeño valor (por ejemplo, 0.01) y un radio de vecindad constante e igual a 1.

No existe un criterio objetivo acerca del número total de iteraciones necesarias para realizar un buen entrenamiento del modelo. Sin embargo, el número de iteraciones debería ser proporcional al número de neuronas del mapa (a más neuronas, son necesarias más iteraciones) e independiente del número de variables de entrada. Aunque 500 iteraciones por neurona es una cifra adecuada, de 50 a 100 suelen ser suficientes para la mayor parte de los problemas (Kohonen, 1990).

3.- Fases en la aplicación de los mapas autoorganizados

En el presente apartado, pasamos a describir las diferentes fases necesarias para la aplicación de los mapas autoorganizados a un problema típico de agrupamiento de patrones.

3.1.- Inicialización de los pesos

Cuando un mapa autoorganizado es diseñado por primera vez, se deben asignar valores a los pesos a partir de los cuales comenzar la etapa de entrenamiento. En general, no existe discusión en este punto y los pesos se inicializan con pequeños valores aleatorios, por ejemplo, entre -1 y 1 ó entre 0 y 1 (Kohonen, 1990), aunque también se pueden inicializar con valores nulos (Martín del Brío y Serrano, 1993) o a partir de una selección aleatoria de patrones de entrenamiento (SPSS Inc., 1997).

3.2.- Entrenamiento de la red

Vista la manera de modificar los vectores de pesos de las neuronas a partir del conjunto de entrenamiento, se van a proporcionar una serie de consejos prácticos acerca de tres parámetros relacionados con el aprendizaje cuyos valores óptimos no pueden conocerse *a priori* dado un problema.

3.2.1.- Medida de similitud

En el apartado 2.3.2. hemos visto las dos medidas de similitud más ampliamente utilizadas a la hora de establecer la neurona vencedora ante la presentación de un patrón de entrada, tanto en la etapa de funcionamiento como en la etapa de aprendizaje de la red. Sin embargo, se debe advertir que el criterio de similitud y la regla de aprendizaje que se utilicen en el algoritmo deben ser métricamente compatibles. Si esto no es así, estaríamos utilizando diferentes métricas para la identificación de la neurona vencedora y para la modificación del vector de pesos asociado, lo que podría causar problemas en el desarrollo del mapa (Demartines y Blayo, 1992).

La distancia euclídea y la regla de aprendizaje presentada son métricamente compatibles y, por tanto, no hay problema. Sin embargo, la correlación o producto escalar y la regla de aprendizaje presentada no son compatibles, ya que dicha regla procede de la métrica euclídea, y la correlación solamente es compatible con esta métrica si se utilizan vectores normalizados (en cuyo caso distancia euclídea y correlación coinciden). Por tanto, si utilizamos la correlación como criterio de similitud, deberíamos utilizar vectores normalizados; mientras que si utilizamos la distancia euclídea, ésto no será

necesario (Martín del Brío y Sanz, 1997). Finalmente, independientemente del criterio de similitud utilizado, se recomienda que el rango de posibles valores de las variables de entrada sea el mismo, por ejemplo, entre -1 y 1 ó entre 0 y 1 (Masters, 1993).

3.2.2.- Tasa de aprendizaje

Como ya se ha comentado, $\alpha(n)$ es la tasa de aprendizaje que determina la magnitud del cambio en los pesos ante la presentación de un patrón de entrada. La tasa de aprendizaje, con un valor inicial entre 0 y 1, por ejemplo, 0.6, decrece con el número de iteraciones (n), de forma que cuando se ha presentado un gran número de veces todo el juego de patrones de aprendizaje, su valor es prácticamente nulo, con lo que la modificación de los pesos es insignificante. Normalmente, la actualización de este parámetro se realiza mediante una de las siguientes funciones (Hilera y Martínez, 1995):

$$\alpha(n) = \frac{1}{n} \qquad \alpha(n) = \alpha_1 \left(1 - \frac{n}{\alpha_2}\right)$$

Siendo α_1 un valor de 0.1 ó 0.2 y α_2 un valor próximo al número total de iteraciones del aprendizaje. Suele tomarse un valor $\alpha_2 = 10000$.

El empleo de una u otra función no influye en exceso en el resultado final.

3.2.3.- Zona de vecindad

La zona de vecindad ($Zona_{j^*}(n)$) es una función que define en cada iteración n si una neurona de salida pertenece o no a la vecindad de la vencedora j*. La vecindad es simétrica y centrada en j*, pudiendo adoptar una forma circular, cuadrada, hexagonal o cualquier otro polígono regular.

En general, $Zona_{j^*}(n)$ decrece a medida que avanza el aprendizaje y depende de un parámetro denominado radio de vecindad R(n), que representa el tamaño de la vecindad actual.

La función de vecindad más simple y utilizada es la de tipo escalón. En este caso, una

neurona j pertenece a la vecindad de la ganadora j* solamente si su distancia es inferior o igual a R(n). Con este tipo de función, las vecindades adquieren una forma (cuadrada, circular, hexagonal, etc.) de bordes nítidos, en torno a la vencedora (figura 3); por lo que en cada iteración únicamente se actualizan las neuronas que distan de la vencedora menos o igual a R(n).

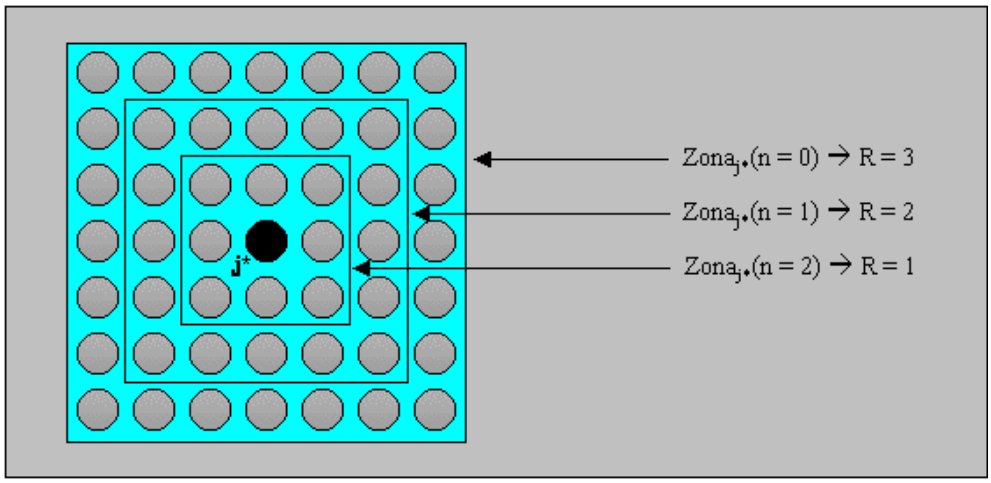


Figura 3: Posible evolución de la zona de vecindad.

También se utilizan a veces funciones gaussianas o en forma de sombrero mejicano (figura 4), continuas y derivables en todos sus puntos, que al delimitar vecindades decrecientes en el dominio espacial establecen niveles de pertenencia en lugar de fronteras nítidas.

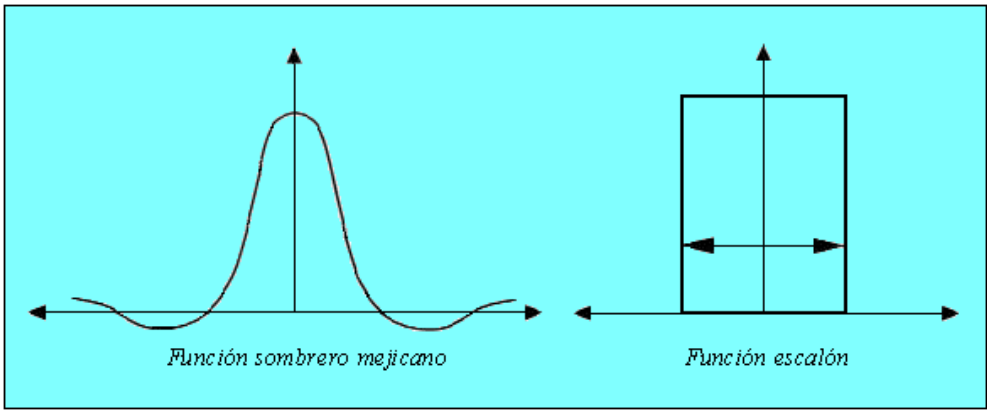


Figura 4: Formas de la función de vecindad.

La función en forma de sombrero mejicano se basa en el tipo de interacción que se produce entre ciertas neuronas del córtex comentado al inicio del documento. Con esta

función, una neurona central emite señales excitatorias a una pequeña vecindad situada a su alrededor. A medida que aumenta la distancia lateral desde la neurona central, el grado de excitación disminuye hasta convertirse en una señal inhibitoria. Finalmente, cuando la distancia es considerablemente grande la neurona central emite una débil señal excitatoria. Por su parte, la función escalón supone una simplificación de la función en forma de sombrero mejicano y, como hemos visto, define de forma discreta la vecindad de neuronas que participan en el aprendizaje.

La zona de vecindad posee una forma definida, pero como hemos visto, su radio varía con el tiempo. Se parte de un valor inicial R_0 grande, por ejemplo, igual al diametro total del mapa (SOM_PAK, 1996; Koski, Alanen, Komu et al., 1996), que determina vecindades amplias, con el fin de lograr la ordenación global del mapa. R(n) disminuye monótonamente con el tiempo, hasta alcanzar un valor final de $R_f = 1$, por el que solamente se actualizan los pesos de la neurona vencedora y las adyacentes. Una posible función de actualización de R(n) es la siguiente (Martín del Brío y Sanz, 1997):

$$R(n) = R_0 + (R_f - R_0)\frac{n}{n_R}$$

donde n es la iteración y n_R el número de iteraciones para alcanzar R_f.

3.3.- Evaluación del ajuste del mapa

En los mapas autoorganizados, el conjunto de vectores de pesos finales va a depender entre otros factores, del valor de los pesos aleatorios iniciales, el valor de la tasa de aprendizaje, el tipo de función de vecindad utilizado y la tasa de reducción de estos dos últimos parámetros. Como es obvio, debe existir un mapa óptimo que represente de forma fiel las relaciones existentes entre el conjunto de patrones de entrenamiento. El mapa más adecuado será aquel cuyos vectores de pesos se ajusten más al conjunto de vectores de entrenamiento. Esto se puede operativizar mediante el cálculo del error cuanficador promedio a partir de la media de $|| X_p - W_{j*}||$, esto es, la media de la diferencia (por ejemplo, la distancia euclídea) entre cada vector de entrenamiento y el vector de pesos asociado a su neurona vencedora (SOM_PAK, 1996). La expresión del error cuantificador promedio utilizada en nuestras simulaciones es la siguiente:

$$Error_{medio} = \frac{\sum_{p=1}^{P}\sum_{i=1}^{N}(x_{pi} - w_{ji})^2}{P}$$

Por tanto, con el objeto de obtener un mapa lo más adecuado posible, deberíamos comenzar el entrenamiento en múltiples ocasiones, cada vez utilizando una configuración de parámetros de aprendizaje diferentes. Así, el mapa que obtenga el error cuantificador promedio más bajo será el seleccionado para pasar a la fase de funcionamiento normal de la red.

3.4.- Visualización y funcionamiento del mapa

Una vez seleccionado el mapa óptimo, podemos pasar a la fase de visualización observando en qué coordenadas del mapa se encuentra la neurona asociada a cada patrón de entrenamiento. Esto nos permite proyectar el espacio multidimensional de entrada en un mapa bidimensional y, en virtud de la similitud entre las neuronas vecinas, observar los clusters o agrupaciones de datos organizados por la propia red. Por este motivo, el modelo de mapa autoorganizado es especialmente útil para establecer relaciones, desconocidas previamente, entre conjuntos de datos.

En la fase de funcionamiento, la red puede actuar como un clasificador de patrones ya que la neurona de salida activada ante una entrada nueva representa la clase a la que pertenece dicha información de entrada. Además, como ante otra entrada parecida se activa la misma neurona de salida, u otra cercana a la anterior, debido a la semejanza entre las clases, se garantiza que las neuronas topológicamente próximas sean sensibles a entradas físicamente similares.

3.5.- Análisis de sensibilidad

Una de las críticas más importantes que se han lanzado contra el uso de RNA trata sobre lo difícil que es comprender la naturaleza de las representaciones internas generadas por la red para responder ante un determinado patrón de entrada. A diferencia de los modelos estadísticos clásicos, no es tan evidente conocer en una red la importancia (o relación) que tiene cada variable de entrada sobre la salida del modelo. Sin embargo,

esta percepción acerca de las RNA como una compleja "caja negra", no es del todo cierta. De hecho, han surgido diferentes intentos por interpretar los pesos de la red neuronal (Garson, 1991; Zurada, Malinowski y Cloete, 1994; Rambhia, Glenny y Hwang, 1999; Hunter, Kennedy, Henry et al., 2000), de los cuales el más ampliamente utilizado es el denominado análisis de sensibilidad.

El análisis de sensibilidad está basado en la medición del efecto que se observa en una salida y_j debido al cambio que se produce en una entrada x_i. Así, cuanto mayor efecto se observe sobre la salida, mayor sensibilidad podemos deducir que presenta respecto a la entrada. En la mayoría de casos, este tipo de análisis ha sido aplicado a la red *backpropagation* (Palmer, Montaño y Calafat, 2000; Palmer, Montaño y Jiménez, en prensa), sin embargo, apenas se han realizado estudios que apliquen el análisis de sensibilidad a los modelos SOM.

En este sentido, Hollmén y Simula (1996) investigaron el efecto de pequeños cambios realizados en una de las variables de entrada sobre la salida de un modelo SOM. Siguiendo un proceso parecido al que se suele aplicar a la red *backpropagation*, estos autores iban realizando pequeños cambios a lo largo de uno de los ejes definido por una variable de entrada (las demás variables se fijaban a un valor promedio) y observando cómo la neurona vencedora iba cambiando de ubicación a lo largo del mapa. Este procedimiento permitió analizar el grado de relación o importancia que tenía cada variable de entrada sobre la salida de la red.

4.- Un ejemplo: Clasificación de la planta del Iris

Como ejemplo ilustrativo vamos a utilizar la matriz de datos sobre el conocido problema de la clasificación de la planta del Iris. Esta matriz proporciona los datos referentes a una muestra de 150 plantas. Cada ejemplar consta de cuatro características y la tarea consiste en determinar el tipo de planta del Iris en base a esas características. Las caracterísitcas son: longitud del sépalo, ancho del sépalo, longitud del pétalo y ancho del pétalo. Hay una submuestra de 50 ejemplares para cada tipo de planta del Iris: Setosa, Versicolor y Virgínica.

El lector interesado puede bajarse esta matriz de datos junto a otras matrices muy conocidas en el ámbito del reconocimiento de patrones (discriminación del cáncer de

mama, detección de cardiopatías, problema OR-Exclusiva, reconocimiento de imágenes via satélite, etc.) en la dirección de Universal Problem Sorvers.

A partir de esta matriz de datos, nos propusimos averiguar si un modelo SOM era capaz de agrupar en el mapa los tres tipos de planta, proporcionándole únicamente los datos sobre las cuatro características citadas. Por tanto, las categorías deberían ser creadas de forma no supervisada por la propia red a través de las correlaciones descubiertas entre los datos de entrada, puesto que no se proporciona la categoría de pertenencia del ejemplar.

Comenzamos con el preprocesamiento de los datos reescalando los cuatro parámetros que iban a servir como variables de entrada a la red. Para ello, se acotó el rango de las variables a valores comprendidos entre 0 y 1. No fue necesario normalizar los vectores de entrada debido a que se utilizaría como criterio de similitud, la distancia euclídea en la etapa de entrenamiento. Como salida de la red se determinó un mapa bidimensional 10x10, por tanto, el mapa estaría compuesto por 100 neuronas de salida.

Se entrenó un total de 10 modelos, cada uno con una configuración inicial de pesos diferente (con rango comprendido entre 0 y 1), pero siguiendo todos el mismo esquema de aprendizaje. Así, el entrenamiento de cada mapa se organizó en dos fases. En la primera fase, cuyo objetivo consiste en organizar el mapa, se utilizó una tasa de aprendizaje alta igual a 1 y un radio de vecindad grande igual al diámetro del mapa, es decir, igual a 10. A medida que avanzaba el aprendizaje, tanto la tasa de aprendizaje como el radio de vecindad iban reduciéndose de forma lineal hasta alcanzar unos valores mínimos, 0.05 y 1, respectivamente. En la segunda fase, cuyo objetivo es el ajuste fino del mapa, se utilizó una tasa de aprendizaje pequeña y constante igual a 0.05, y un radio de vecindad constante y mínimo igual a 1. La primera fase constó de 1000 iteraciones, mientras que la segunda fase constó de 2000 iteraciones.

Una vez entrenados los 10 modelos, se calculó para cada uno de ellos, el error cuantificador promedio. Quedó seleccionado el modelo cuyo error fue el más pequeño --con un valor igual a 0.0156.

A continuación, se muestra el mapa del modelo finalmente seleccionado (figura 5):

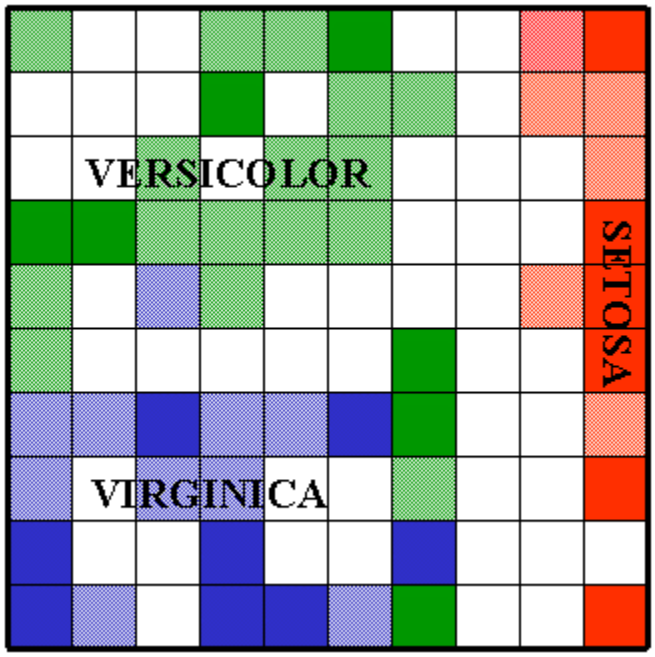


Figura 5: Mapa del modelo seleccionado.

En la figura, las neuronas de salida que representan ejemplares de planta Versicolor aparecen de color verde, las neuronas que representan ejemplares de planta Virgínica aparecen de color azul y, finalmente, las neuronas que representan ejemplares de planta Setosa aparecen de color rojo. Las neuronas de color blanco no representan patrón alguno. Con el objeto de poder analizar la distribución de frecuencias de cada clase de planta en el mapa, cada color puede aparecer con dos posibles intensidades según la frecuencia de patrones asociados a una determinada neurona de salida. Así, cuando el color es poco intenso, hay pocos patrones (1 ó 2) asociados a la neurona; cuando el color es intenso, hay muchos patrones (de 3 a 10) asociados a la neurona.

Se puede observar que los ejemplares de planta Setosa han sido perfectamente discriminados respecto a las otras dos categorías, quedando agrupados todos en la parte derecha del mapa. Por su parte, los ejemplares de Versicolor y Virgínica se reparten la parte superior e inferior izquierda del mapa, respectivamente; compartiendo zonas adyacentes. Esto parece indicar que la planta Setosa se diferencia perfectamente de los otros dos tipos de planta, mientras que la planta Versicolor y Virgínica mantienen características más similares, aunque bien diferenciables.

A continuación, podríamos pasar a la etapa de funcionamiento de la red donde se presentarían nuevos ejemplares y, mediante el proceso de competición, podríamos

observar en que zona del mapa está ubicada la neurona de salida vencedora asociada a cada nuevo ejemplar.

5.- Recursos gratuitos en internet sobre los mapas autoorganizados

En la Web podemos encontrar multitud de recursos relacionados con el campo de las RNA. A modo de ejemplos ilustrativos, describiremos el funcionamiento de dos applets y un programa totalmente gratuitos que permiten simular el comportamiento de un mapa autoorganizado de Kohonen tal como ha sido expuesto a lo largo de este documento.

5.1.- Applets

Se han seleccionado dos ejemplos de applets, creados por Jussi Hynninen del Laboratorio Tecnológico de Acústica y Procesamiento de Señales de Audio de la Universidad de Helsinki (Finlandia) y colaborador de Kohonen. Uno simula un mapa autoorganizado unidimensional, mientras que el otro simula un mapa autoorganizado bidimensional. El lector interesado en este tipo de recursos puede visitar la página del Instituto Nacional de Biociencia y Tecnología Humana (M.I.T.I.) de Japón, donde podrá encontrar un numeroso listado de applets demostrativos sobre RNA.

5.1.1.- Ejemplo de mapa autoorganizado unidimensional

En el primer applet (figura 6) se representa un mapa unidimensional mediante una fila de cinco barras verticales en forma de termómetro. Estas barras representan el valor (un valor escalar) de las neuronas de salida del mapa y están controladas por el algoritmo del modelo SOM. Cuando se presentan patrones de entrada aleatorios consistentes en un valor escalar, los valores de las barras se van organizando gradualmente. Durante el entrenamiento, la diferencia entre el nuevo y el anterior valor de una barra se muestra de color amarillo después de la presentación de un patrón. La barra cuyo valor es el más cercano al patrón de entrada (es decir, la barra que representa la neurona vencedora), aparece de color rojo. El tamaño de la zona de vecindad es constante y con un radio igual a 1, es decir, solo se modifica el valor de la neurona vencedora y el de sus adyacentes. La barra de color rosado situada en el extremo izquierdo es la que controla

el patrón de entrada, ya que muestra el valor de la entrada presentada al mapa en cada paso del entrenamiento.

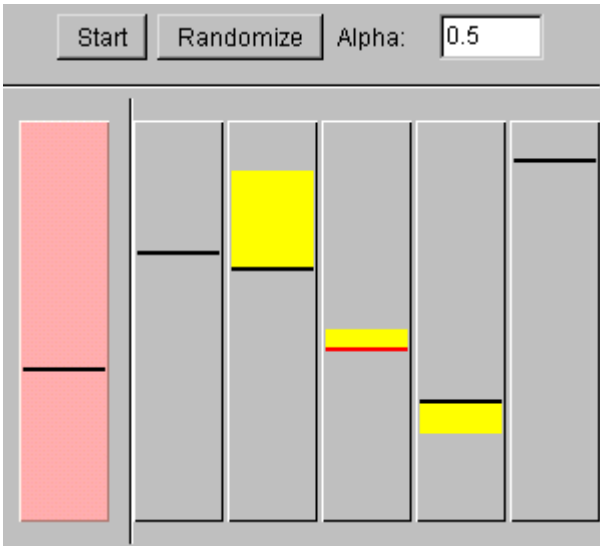


Figura 6: Applet demostrativo de un mapa unidimensional.

El mapa puede ser entrenado de forma automática o manual. Durante el entrenamiento automático, se presentan patrones de entrada aleatorios. En el entrenamiento manual el usuario puede especificar el patrón de entrada que será utilizado en el aprendizaje, simplemente haciendo click con el ratón en el controlador de entrada (la barra de color rosado). Presionando la tecla ESPACIO, se realizará un ciclo de aprendizaje con un patrón de entrada aleatorio.

Usando el botón START/STOP situado en el applet, comenzará o parará el entrenamiento automático. Por su parte, el botón RANDOMIZE inicializará los valores de las neuronas con valores aleatorios. Se puede usar el campo de texto *Alpha* para determinar el valor de la tasa de aprendizaje, escribiendo el valor y luego pulsando la tecla ENTER. Finalmente, el valor de las neuronas del mapa también se pueden modificar haciendo click con el ratón sobre las barras verticales.

5.1.2.- Ejemplo de mapa autoorganizado bidimensional

En el segundo applet se representa un mapa bidimensional (figura 7) formado por un rectángulo de 4x5 neuronas.

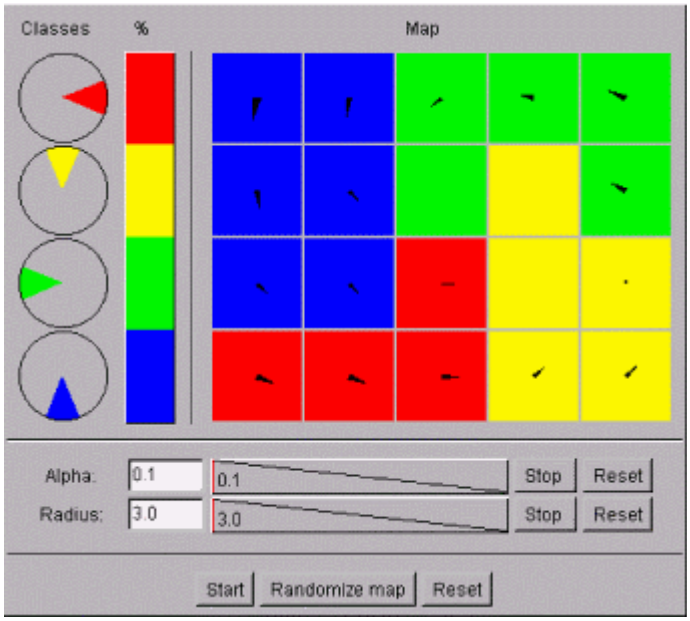


Figura 7: Applet demostrativo de un mapa bidimensional.

En este applet, los cuatro gráficos de "pastel" situados en la parte izquierda definen las clases o categorías de entrada. Hay cuatro clases representadas por los colores rojo, amarillo, verde y azul. En cada gráfico, el sector dibujado de color representa la distribución de vectores (formados por dos valores: x y) que la clase produce. Los cuatro gráficos proporcionan vectores aleatorios que se distribuyen en el sector mostrado en el gráfico. Los gráficos puede ser editados con el ratón de dos maneras diferentes: haciendo click cerca del centro del gráfico podemos cambiar la ubicación del sector, haciendo click cerca del borde del gráfico podemos cambiar la anchura del sector.

La barra vertical situada a la derecha de los gráficos de "pastel" determina las proporciones relativas de las cuatro clases respecto a los datos de entrada. Inicialmente los vectores de entrada se distribuyen de forma igualitaria entre las cuatro clases. El usuario puede cambiar las proporciones con el ratón: haciendo click en el borde que delimita dos clases y, a continuación, subiendo o bajando el cursor.

El área de la derecha del applet corresponde al mapa autoorganizado. El mapa consiste en 20 neuronas organizadas en un rectángulo de 4x5 neuronas. Durante el entrenamiento, el color de fondo de las neuronas corresponde al color de la clase del patrón de entrada más próximo.

A cada neurona del mapa le corresponde un vector de referencia de dos dimensiones cuyos componentes se interpretan como dos coordenadas, x e y. Cada unidad del mapa se representa mediante una flecha que apunta desde las coordenadas [0, 0] a las coordenadas [x, y] almacenadas en el vector de referencia. El rango de las coordenadas oscila entre -1 y 1.

Debajo de los gráficos de "pastel" y del mapa está el panel de control que determina el valor de la tasa de aprendizaje (alfa) y el radio de vecindad. Ambos parámetros funcionan de la misma forma: en la parte izquierda se sitúa el campo de texto donde el usuario puede introducir el valor que desee del parámetro y, a continuación, apretar la tecla ENTER. Por defecto, el valor de la tasa de aprendizaje alfa se reduce durante el aprendizaje desde el valor inicial hasta el valor 0 y el valor del radio de vecindad decrece desde el valor inicial hasta el valor 1. El usuario puede determinar un valor constante para estos parámetros, pulsando la tecla STOP situada a la derecha del panel que controla el parámetro correspondiente. Para que sigua decreciendo, pulse otra vez el botón (leerá START). Pulsando el botón RESET reinicia el valor del parámetro a su valor inicial.

Finalmente, el panel de control situado en la parte inferior del applet se utiliza para iniciar y parar el aprendizaje, y también para iniciar con valores aleatorios los vectores de pesos del mapa. Pulse el botón START para comenzar el aprendizaje del mapa, pulse otra vez para parar. Pulsando el botón RANDOMIZE MAP inicializa el mapa con valores aleatorios. El botón RESET permite reiniciar los parámetros tasa de aprendizaje alfa y radio de vecindad.

5.2.- Software

En la Web podemos encontrar multitud de programas simuladores de redes neuronales de libre distribución (gratuitos o *sharewares*). El lector interesado puede visitar dos listados completos sobre este tipo de programas: Por un lado, tenemos el listado ofrecido por el grupo de noticias sobre redes neuronales comp.ai.neural-nets, por otro lado, tenemos el listado ofrecido por el Pacific Northwest National Laboratory.

5.2.1.- SOM_PAK

De entre los simuladores de libre distribución que podemos encontrar en internet, cabe destacar el programa SOM_PAK (para UNIX y MS-DOS), software de dominio público desarrollado en la Universidad de Tecnología de Helsinki (Finlandia) por el grupo de Kohonen para el trabajo y experimentación con el modelo SOM.

Para bajarse el programa SOM_PAK (versión 3.1), pulse el siguiente botón:

En este mismo enlace, se puede obtener el manual de usuario del programa (SOM_PAK, 1996), por tanto, remitimos al lector la lectura del manual para una descripción del funcionamiento del programa. Este sencillo manual proporciona numerosos consejos prácticos y, mediante un sencillo ejemplo, el usuario puede seguir la secuencia de pasos necesarios (inicialización, entrenamiento, evaluación y visualización del mapa) para la construcción de un modelo SOM.

Referencias bibliográficas

Arbib, M.A., Erdi, P. y Szentagothai, J. (1997). *Neural organization: structure, function and dynamics*. Cambridge, Mass.: MIT Press.

Demartines, P. y Blayo, F. (1992). Kohonen self-organizing maps: Is the normalization necessary?. *Complex Systems, 6,* 105-123.

Garson, G.D. (1991). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.

Hilera, J.R. y Martínez, V.J. (1995). *Redes neuronales artificiales: Fundamentos, modelos y aplicaciones*. Madrid: Ra-Ma.

Hollmén, J. y Simula, O. (1996). Prediction models and sensitivity analysis of industrial process parameters by using the self-organizing map. *Proceedings of IEEE Nordic Signal Processing Symposium (NORSIG'96)*, 79-82.

Hunter, A., Kennedy, L., Henry, J. y Ferguson, I. (2000). Application of neural networks and sensitivity analysis to improved prediction of trauma survival. *Computer Methods and Programs in Biomedicine, 62*, 11-19.

Kohonen, T. (1982a). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59-69.

Kohonen, T. (1982b). Analysis of a simple self-organizing process. *Biological Cybernetics, 44*, 135-140.

Kohonen, T. (1989). *Self-organization and associative memory*. New York: Springer-Verlag.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE, 78*(9), 1464-1480.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer-Verlag.

Koski, A., Alanen, A., Komu, M., Nyman, S., Heinänen, K. y Forsström, J. (1996). Visualizing nuclear magnetic resonance spectra with self-organizing neural networks. En: Taylor, J.G. (Ed.). *Neural networks and their applications* (pp. 83-92). Chichester: John Wiley and Sons.

Martín del Brío, B. y Sanz, A. (1997). *Redes neuronales y sistemas borrosos*. Madrid: Ra-Ma.

Martín del Brío, B. y Serrano, C. (1993). Self-organizing neural networks for analysis and representation of data: some financial cases. *Neural Computing and Applications, 1*, 193-206.

Masters, T. (1993). *Practical neural networks recipes in C++*. London: Academic Press.

Palmer, A., Montaño, J.J. y Calafat, A. (2000). Predicción del consumo de éxtasis a partir de redes neuronales artificiales. *Adicciones, 12*(1), 29-41.

Palmer, A., Montaño, J.J. y Jiménez, R. (en prensa). Tutorial sobre redes neuronales artificiales: el perceptrón multicapa. *Internet y Salud*, URL: http://www.intersalud.net

Rambhia, A.H., Glenny, R. y Hwang, J. (1999). Critical input data channels selection for progressive work exercise test by neural network sensitivity analysis. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1097-1100.

Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Sarle, W.S. (Ed.) (1998). *Neural network FAQ*. Periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: ftp://ftp.sas.com/pub/neural/FAQ.html.

SOM_PAK (1996). *The Self-Organizing Map Program Package: User's Guide* [Manual de programa para ordenadores]. Helsinki University of Technology, Finland.

SPSS Inc. (1997). *Neural Connection 2.0: User´s Guide* [Manual de programa para ordenadores]. Chicago: SPSS Inc.

Taylor, J.G. (1996). *Neural networks and their applications*. Chichester: John Wiley and Sons.

Zurada, J.M., Malinowski, A. y Cloete, I. (1994). Sensitivity analysis for minimization of input data dimension for feedforward neural network. *Proceedings of IEEE International Symposium on Circuits and Systems*, 447-450.

Accesible en Internet desde el 25/4/2001
http://www.psiquiatria.com/psicologia/revista/67/3301

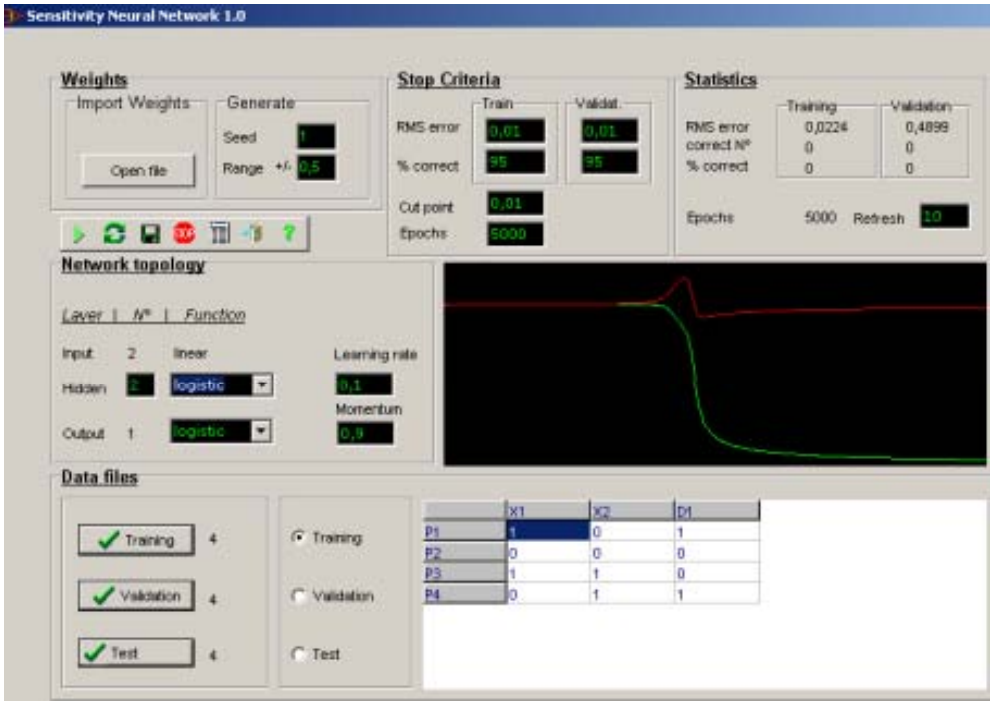
Anexo 2: Sensitivity Neural Network 1.0: User's Guide

General information

Overview

Artificial Neural Networks (ANN) of the multilayer perceptron type associated with the backpropagation error algorithm are the most widely used model of networks in the field for the estimation of continuous variables and the pattern classification. The strength and flexibility of this computational technology contrast with its apparent incapacity to quantify the importance or the effect that each explicative variable has on the prediction made by the model. In the last years diverse methodologies – generally called sensitivity analysis – directed at overcoming this limitation have been proposed, obtaining promising results. However, current commercial and free distribution software programs of ANN have not implemented these methodologies.

Sensitivity Neural Network 1.0 (SNN) is an easy to use program that permits the simulation of the behavior of a multilayer perceptron network (input layer, hidden layer and output layer) trained by means of the backpropagation algorithm, and implements a set of sensitivity methods that have been demonstrated in scientific literature to be effective in the measurement of the input variables effect on the neural network output.



What are Artificial Neural Networks?

In this section we will introduce the reader to the field of ANN and briefly explain the mathematical algorithms of ANN implemented in SNN. To obtain more detailed information, please consult the references.

Introduction

ANN are information processing systems whose structure and function are inspired by biological neural networks. They consist of a large number of simple processing elements called nodes or neurons that are organized in layers. Each neuron is connected with other neurons by connection links, each of which is associated to a weight. The knowledge that the ANN has about a given problem is found in these weights.

The use of ANN can be directed in two ways: either as models for the study of the nervous system and cognitive phenomena, or as tools for the resolution of practical problems such as the pattern classification and the approximation of functions. From this second perspective, ANN have been satisfactorily applied in the prediction of diverse problems in different areas of study – biology, medicine, economy, psychology, etc. - obtaining excellent results compared to models derived from classic statistics. The advantage of ANN is its capacity to learn complex functions, or non-linears, between variables without the necessity of applying suppositions or restrictions a priori to the data.

As adaptive systems, ANN learn from experience, that is they learn to carry out certain tasks through training with illustrative examples. By means of this training or learning, ANN create their own internal representation of the problem, for which they are said to be self-organized. Later, they can respond adequately when they are presented with situations to which they have not been previously exposed to; in other words, ANN are capable of generalizing from previous cases to new cases. This characteristic is fundamental, as it permits the solution of problems in which distorted or incomplete information has been presented.

In summary, it can be said that ANN imitate the structure of the nervous system, with the intention of constructing parallel, distributed, adaptive information processing systems that present some intelligent behavior.

Multilayer perceptron and the backpropagation algorithm

Multilayer perceptron

A multilayer perceptron is composed of an input layer, an output layer, and one or more hidden layers, although it has been demonstrated that for the majority of problems only one hidden layer is enough. In figure 1 we can observe a typical perceptron formed by an input layer, a hidden layer and an output layer.

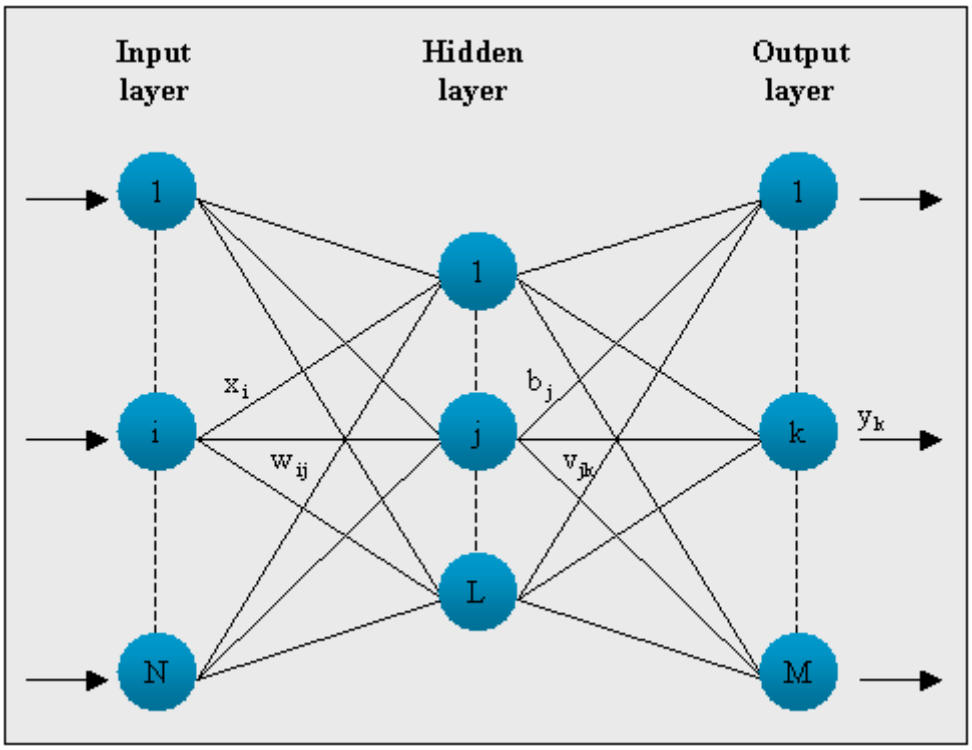


Figure 1: Perceptron composed of three layers.

In this type of architecture the connections between neurons are always towards the front. In other words, the connections go from the neurons of a certain layer towards the neurons of the following layer; there are no lateral connections or connections towards the back. Therefore the information is always transmitted from the input layer towards the output layer. The threshold of hidden neurons and the output neurons is considered as a weight associated with a dummy neuron with an output value equal to 1. We can consider *wij* to be the connection weight between the input neuron *i* and the hidden neuron *j*, *vjk* to be the connection weight between the hidden neuron *j* and the output neuron *k*.

The backpropagation algorithm

In the backpropagation algorithm (Rumelhart, Hinton, and Williams, 1986) we can consider on the one hand, a stage of functioning where we see, with the trained network, an input pattern composed of the values of explicative variables for a given register, which is transmitted through various layers of neurons until obtaining an output, and on the other hand, a stage of training or learning where the weights of the network are modified in such a way that the output desired by the user coincides with the output obtained by the network with the presentation of a determined input pattern.

Stage of functioning: When we present a pattern *p* of input *Xp: xp1,...,xpi,... xpN*, this is transmitted through the weights *wij* from the input layer towards the hidden layer. The neurons of this intermediate layer transform the signals received by means of the application of an activation function *f(.)* providing, thus, an output value *bj*. This is transmitted through the weights *vjk* towards the output layer, where the same operation is applied as in the previous case, and the neurons of this layer provide the outputs *yk* of the network. As an activation function of the hidden neurons and the output neurons the linear function, the logistic sigmoid and the hyperbolic tangent sigmoid are normally used (See figure 2).

Stage of learning: The main goal of this stage is to minimize the discrepancy or error between the output obtained by the network and the output desired by the user with the presentation of a set of patterns called the training group. This goal is reached by modifying the weights in an iterative manner by means of the following expressions:

$$\Delta w_{ij}(n+1) = \eta\left(\sum_{p=1}^{P} \delta_{pj} x_{pi}\right) + \alpha\, \Delta w_{ij}(n) \qquad \Delta v_{jk}(n+1) = \eta\left(\sum_{p=1}^{P} \delta_{pk} b_{pj}\right) + \alpha\, \Delta v_{jk}(n)$$

for a connection weight between an input neuron *i* and a hidden neuron *j*, and for a connection weight between a hidden neuron *j* and an output neuron *k*, respectively. The values of eta and alfa represent the learning rate and the momentum factor respectively. Both elements control the size of the change of weights in each epoch. Finally, the delta values represent the contribution of the neuron to the error committed in the output of the network.

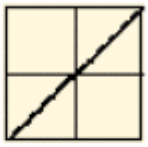
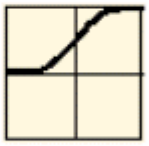
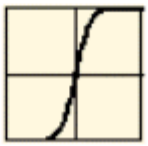
Name	Function $y = f(x)$	
Linear	x	
Logistic	$\frac{1}{1+e^{-x}}$	
Tanh	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	

Figure 2: The activation functions most frequently used in the backpropagation network.

Analysis of the input variables effect

There are two types of general methodologies which permit us to know what the network has learned from the weight values and the activation values, that is, what we are aiming at is to know the effect or importance of each input variable on the network output. These two methodologies are: analysis based on the magnitude of weights and sensitivity analysis.

Analysis based on the magnitude of weights

The analysis based on the magnitude of weights groups together those procedures that are based exclusively on the values stored in the static matrix of weights for the purpose of determining the relative influence of each input variable on each one of the outputs of the network. One of the equations most often used based on the magnitude of weights is the one proposed by Garson (1991) that states:

$$Q_{ik} = \frac{\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)}{\sum_{i=1}^{N}\left(\sum_{j=1}^{L}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)\right)}$$

where $\sum_{r=1}^{N} w_{rj}$ is the sum of the connection weights between the input neurons *i* and the hidden neuron *j*. In this equation we should take into account that, on the one hand, the value of the weights is taken in absolute value so that the positive and negative weights don't cancel each other out, and on the other hand, the threshold value of the hidden neurons and the output neurons are not taken into account, assuming that their inclusion does not affect the final result. The index *Qik* represents the percentage of influence of the input variable *i* on the output *k*, in relation to the rest of the input variables, in such a way that the sum of this index for all input variables should give a value of 100%.

Sensitivity analysis

The sensitivity analysis is based on the measurement of the effect that is observed in the output *yk* due to the change that is produced in the input *xi*. Thus, the greater the effect observed on the output, the greater the sensitivity we can deduce that will be present with respect to the input. In the following section we present two approaches to the sensitivity analysis: the Jacobian sensitivity matrix and the numeric sensitivity method.

Jacobian sensitivity matrix: The elements that make up the Jacobian matrix S provide, analytically, a sensitivity measurement of the output to changes that are produced in each one of the input variables. In the Jacobian matrix S - of the order *NxM* – each row represents an input in the network, and each column represents an output in the network, in such a way that the element *Sik* of the matrix represents the sensitivity of the output *k* with respect to the input *i*. Each of the *Sik* elements is obtained by calculating the partial derivative of an output *yk* with respect to an input *xi*, that is $\partial y_k / \partial x_i$ (Bishop, 1995). In this case the partial derivative represents the instant slope of the underlying function between *xi* and *yk* for some values given in both variables. The greater the absolute value of *Sik*, the more important the *xi* in relation to *yk*. The

sign of *Sik* indicates whether the change observed in *yk* is going in the same direction or not as the change provoked in *xi*. Since different input patterns can give different slope values, the sensitivity needs to be evaluated from the whole of the training set. In this way, it is useful to obtain a series of summary indexes such as the arithmetic mean, the standard deviation, the mean square, the minimum and maximum value of sensitivity.

Numeric sensitivity method: This method presented recently by our team is based on the computation of slopes that are formed between inputs and outputs, without making any suppositions about the nature of the variables. This method is especially useful in those cases in which discrete variables are being handled (for example, gender: 0 = male, 1 = female or status: 0 = healthy, 1 = ill), due to the fact that the computation of slopes by means of a partial derivative is based on the supposition that all the variables implied in the model are of a continuous nature. This method consists of placing in ascending order the values of the input variable of interest, and in function of this ordering, group together the registers in various groups: 30 groups when the variable is continuous and two groups when the variable is discrete binari (in this case, one group when the variable takes the minimum value (for example, 0), the other group when the variable takes the maximum value (for example, 1). The computation of the slope that is formed between the input variable *xi* and the output variable *yk* for each pair of consecutive groups allows us to take into account the effect of the input on the output. As in the computation of the Jacobian sensitivity matrix, with the aim of summarizing the information, we obtain as summary indexes the arithmetic mean, the standard deviation, the mean square, the minimum and maximum of the slope.

System Requirements

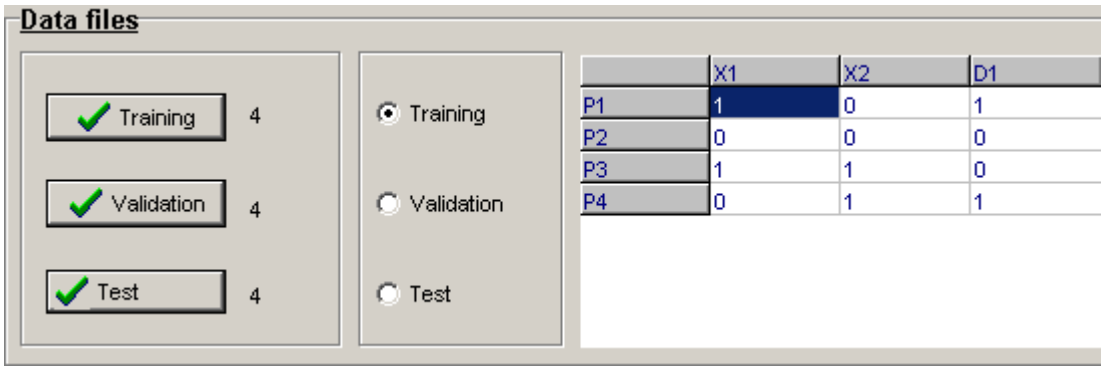
SNN requires an IBM PC or compatible computer with a 486 processor or greater, 16 MB of RAM memory and Windows operating system. The possibility of handling large size data matrices and complex neural architectures depends on the potency of the computer used.

The complete installation of SNN requires less than 5 MB of disk space.

Principal window

Data files

Allows us to open and visualize the data matrices necessary to train and validate a neural network.



By clicking on Training, Validation, or Test, a dialogue box appears that permits us to select the file that contains the data that will be used in the training, validation, and testing, respectively.

Once the data matrices have been activated, they can be alternatively visualized in a simple spreadsheet by clicking on the corresponding radial button.

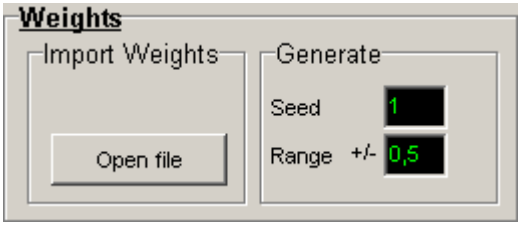
The files that contain the data matrices should be edited in ASCII format with TXT extension and headed by a parenthesis for the number of patterns (P), the number of input variables (Xi) and the number of output variables (Dk). We present as an example the structure of the data file xor.txt which is made up of four patterns, two input variables and one output variable:

(4 2 1)

1,00 0,00 1,00
0,00 0,00 0,00
1,00 1,00 0,00
0,00 1,00 1,00

Weights

This allows us to randomly initialize the weights of the network or import the weights trained in a previous session.



The connection weights and the threshold weights of the network can be initialized by means of an random process in the option Generate from a seed value and a range of determined values.

The weights can be imported from a file previously saved either by SNN or by another ANN simulator program. To do so, click Open file from the option Import and select the file that contains the value of the saved weights.

The file of weights, in ASCII format with TXT extension, should be headed by the number of input neurons, the number of hidden neurons and the number of output neurons. The weights are organized in a single vector row in the following way: matrix (NxL) of connection weights between input and hidden neurons; vector (1xL) of threshold weights of hidden neurons; matrix (LxM) of connection weights between hidden neurons and output neurons; vector (1xM) of threshold weights of the output neurons. In the second and third line of the file the activation function should appear (linear, logistic, or tanh) of the hidden neurons and the output neurons, respectively. Then, the content of a file of weights saved previously is shown, called weights.txt, that resolves the XOR problem:

N=2 L=2 M=1 -6,17 6,66 5,91 -6,70 ;-3,26 -3,71 ;10,21 10,04 ;-5,00
logistic
logistic

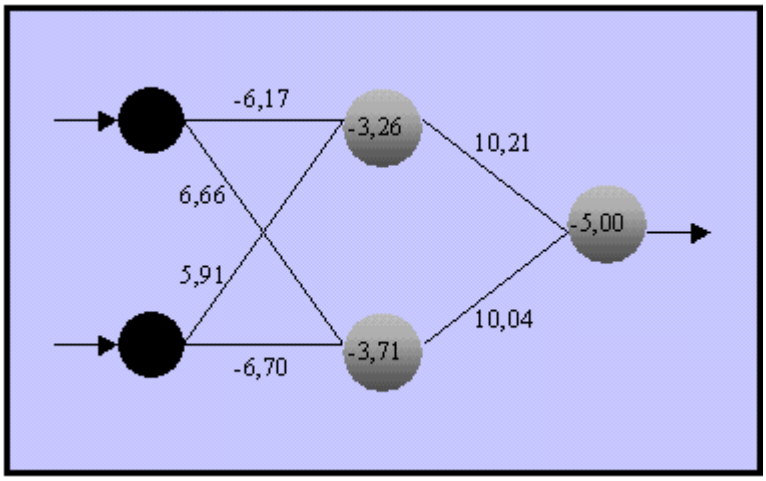
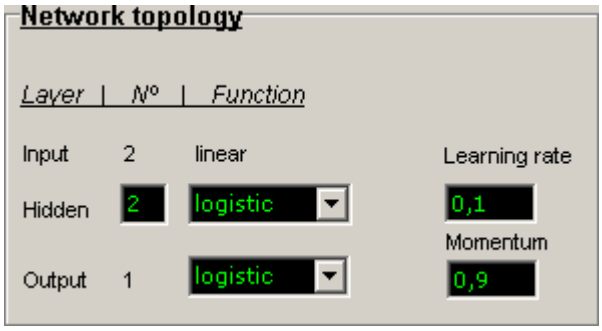


Figure 3: Reconstruction of the weights saved in the file weights.txt.

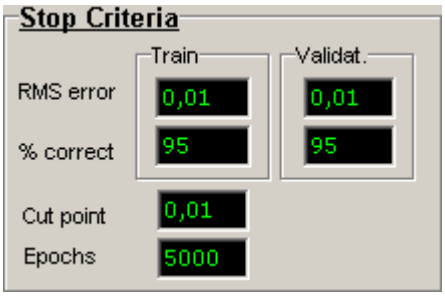
Network Topology

This allows us to determine the number of neurons of the hidden layer, the activation functions (linear, logistic sigmoid, and hyperbolic tangent sigmoid) of the hidden neurons and the output neurons, as well as the value of the learning rate (eta) and the value of the momentum factor (alfa).



Stop Criteria

Allows us to stop automatically the training of the network, in function of the completion of a series of criteria.



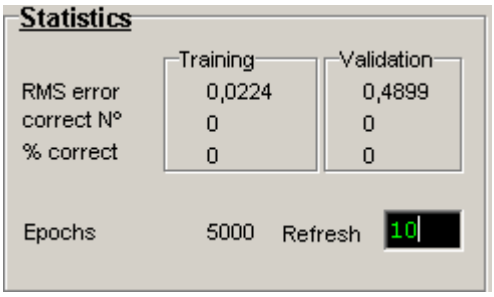
- *RMS error in training and validation*: The training will stop when the RMS error (Square root of the Mean Square of error) of the training group and the validation group is equal or inferior to the value specified in the corresponding boxes.

- *% correct*: The training will stop when the percentage of patterns correctly classified in the training group and validation group is equal or superior to the specified value. We consider that a pattern has been correctly classified when the RMS error committed by the network for this pattern is inferior to the determined cutting off point.

- *Epochs*: The training will stop when the number of specified epochs has been reached, independently of whether or not any of the previous stop criteria have been completed.

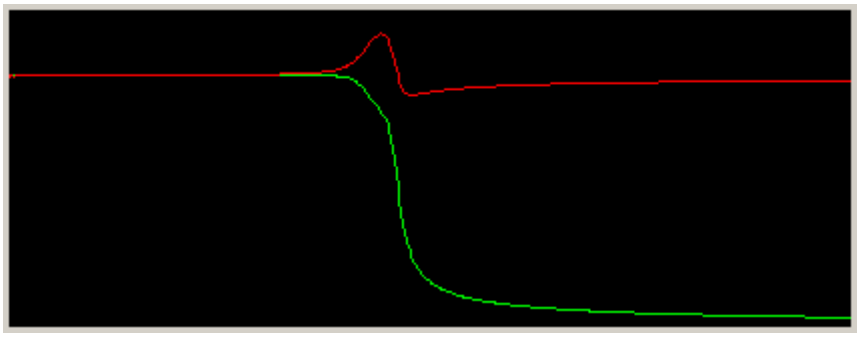
Statistics

During the process of learning of a neural network, SNN provides a series of statistics of interest: the RMS error, the number and percentage of patterns correctly classified and, finally, the number of epochs made in each instant.



A graphic representation is also provided of the evolution of the RMS error for the training group and validation group as the number of epochs increases. The green line represents the RMS error of the training group, the red line represents the RMS error of the validation group. The field Refresh allows us to determine every how many epochs

we want the graphics to be updated, this being possible even during the process of learning.



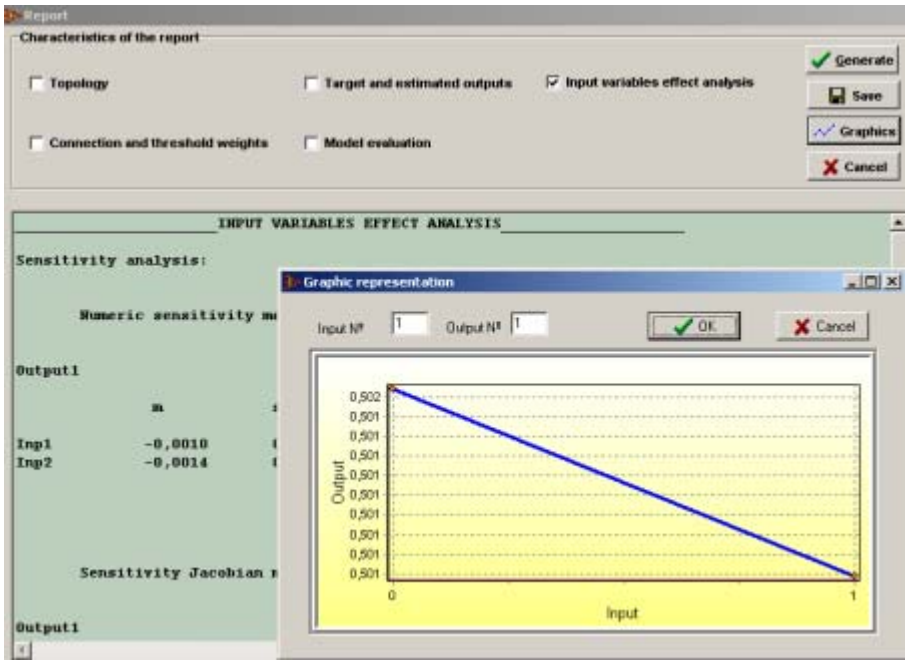
Action buttons

The action buttons permit us to begin the training, cleaning configuration, saving of the trained weights in the format described under the section Weights, stop the training, generate a report of the results, exit the program, and consult the help file, respectively.

Report window

Characteristics of the report

Once a configuration of weights has been obtained by means of training the network in the current session, or by importing a file of weights saved in a previous session, we can generate a report of the results. By clicking on the action button Generate report, a new window is opened in which we can select the characteristics of the report. This is made on the data matrix (training, validation or test) that is activated in that moment in the principal window by means of the corresponding radial button. By clicking on Generate the results of the options selected will appear on the screen. Also, the button Save allows us to save the content of the report in a file with extension txt.



The different elements that can make up the report are:

- *Topology*: For each layer of neurons (input, hidden and output), the number of neurons and the type of activation function used is provided. The function of the input layer is always linear.

- *Connection weights and threshold weights*: This facilitates the value of the connection weights and threshold weights of the network in a matrix format: the matrix NxL of connection weights between the N input neurons and the L hidden neurons, the vector 1xL of threshold weights of the hidden neurons, the matrix LxM of connection weights between the L hidden neurons and the M output neurons and the vector 1xM of threshold weights of the output neurons.

- *Target outputs and estimated outputs*: This provides for each pattern the value of the input variables, the output desired by the user (target) and the output calculated by the neural network.

- *Model evaluation*: This allows us to obtain the RMS error for each output neuron, the total RMS error of the neural network and the number of patterns correctly classified in function of the value of the cutting off point established in the principal window.

- *Input variables effect analysis*: This carries out the numeric sensitivity analysis and the analytic sensitivity analysis (Jacobian sensitivity matrix), providing for each pair of

input-output variables: the arithmetic mean (m), the standard deviation (sd), the mean square (ms), the minimum value (min) and the maximum value (max). It also carries out the analysis based on the magnitude of weights by Garson's method. Once the input variables effect analysis has been done, the Graphics button is activated, which allows us to produce the graphic representation of the function learned by the network between each pair of input-output variables, obtained by means of the numeric sensitivity analysis.

Examples of data matrices

Two examples of data matrices have been proposed that can be used to test the functioning of the SNN program:

- Classification of the plant iris: The matrix iris.txt contains the data on the known problem of the classification of the plant iris. It consists of a sample of 150 plants. Each specimen has four characteristics, and the task consists of determining the type of iris plant based on these characteristics. The characteristics are: length of the sepal, width of the sepal, length of the petal and width of the petal. There is a sub-sample of 50 specimens for each type of iris: Setosa. Versicolor and Virginica. The four characteristics act as input variables and the type of plant acts as an output variable with the following codification: Setosa (1 0 0), Versicolor (0 1 0) and Virginica (0 0 1). The file irisweights.txt contains a configuration of weights adapted to this data matrix.

- XOR problem: This consists of discriminating between two classes of patterns in a bi-dimensional space (X,Y). Thus, the input patterns (0, 0) and (1, 1) belong to class A (output = 0) and the input patterns (1, 0) and (0, 1) belong to class B (output = 1). The matrix xortrain.txt contains the data of the XOR problem. The matrices xorvalida.txt and xortest.txt contain the XOR problem with the introduction of noise in the output. It could be interesting to use these two matrices as validation data and test data respectively, in order to analyze the yield of the network with degraded data. The file xorweights.txt contains a configuration of weights adapted to the XOR problem.

References

Bishop, C.M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. En: D.E. Rumelhart & J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press.

Garson, G.D. (1991). Interpreting neural-network connection weights. *AI Expert*, April, 47-51.